Universität - Gesamthochschule - Paderborn
Fachbereich Wirtschaftswissenschaften
Lehrstuhl für Wirtschaftsinformatik 2

D I P L O M - A R B E I T

Neue Konzepte zur computergestützten Organisationsmodellierung:

Untersuchung und Konzeption eines computergestützten
Hilfmittels zur Organisationsmodellierung zum Einsatz in
innovativen Workflow Management Systemen
(Organization Object Modeler)

zur Erlangung des Grades eines Diplom-Kaufmanns

vorgelegt bei

Prof. Dr. Ludwig Nastansky

betreut von
Dipl. Wirtsch.-Inf. Marcus Ott

Stefan Meyer
Paul-Michels-Weg 4
33100 Paderborn

SOMMERSEMESTER 1996

INHALTSVERZEICHNIS

Kapitel 1: Einleitung

Die marktwirtschaftlich orientierten Unternehmen sind zunehmend verschärften Wettbewerbsbedingungen ausgesetzt. Im international immer homogeneren Wirtschaftsgeschehen müssen sie auf komplexe, sich rasch verändernde Umwelteinflüsse in Wirtschaft, Technik, Politik und Gesellschaft reagieren. Durch die konjunkturelle Entwicklung verbunden mit vielfältiger Konkurrenz und in manchen Bereichen ausgeschöpften Absatzmärkten, sind manche sogar in eine ökonomisch prekäre Situation geraten.

Da sie sich möglichst schneller als die Konkurrenz an zunehmende Umweltdynamik und Umweltkomplexität anpassen müssen, sind bewährte Eigenschaften hierarchischer Unternehmensformen wie Zuverlässigkeit, Reibungslosigkeit und Schnelligkeit der Funktionserfüllung sowie scheinbare Konfliktfreiheit keine Erfolgsgaranten mehr. Andersartige Eigenschaften bzw. Fähigkeiten wie Kreativität, Flexibilität, hohes Informationsverarbeitungsvermögen, Innovationsfähigkeit und höheres differenziertes Selbstregulationspotential treten deshalb in den Vordergrund[1]. Eine sich permanent wandelnde Umwelt hat zu steigenden Unternehmensgrößen, steigender Unternehmenskomplexität und Unternehmensdifferenzierung geführt. Das bedingt zugleich eine fortschreitende Arbeitsteilung, Aufgabendelegation, damit verbunden eine Spezialisierung der Funktionen, und stellt höhere Anforderungen an die Selbstkoordinationsfähigkeit der sozialen Systeme[2].

Ein möglicher Weg, um den Produktions- oder Wertschöpfungsprozeß eines Unternehmens effizienter zu gestalten, ist der Bereich der Büroautomation. „Neuzeitliche Büroausstattungen, verbunden mit einem hohen Grad der Automatisierung prägen Unternehmensorganisationen. Vergleicht man jedoch die Automatisierung im Büro- und Verwaltungsbereich mit dem Grad der Automatisierung in den Produktionsstätten der meisten Fertigungsbetriebe, dann wird ein Nachholbedarf im Büro- und Verwaltungsbereich sichtbar."[3] Bisher hat eine rapide anwachsende Infrastruktur im Bereich der Bürokommunikation eine Abkehr von reiner Kommunikation zur Büroautomatisierung bewirkt. Automatisch dirigierte und kontrollierte Geschäftsprozesse sind in vielen Verwaltungen und Forschungseinrichtungen alltäglich geworden. Ein Wegbereiter und vielleicht das innovativste System dieser Entwicklung ist das Produkt *Notes* der Firma Lotus, das die technischen Voraussetzungen für diese Arbeit liefert.

Es hat sich herausgestellt, daß eine flexible Struktur in den Geschäftsprozessen, die sog. **Ablauforganisation**, durch ein Festhalten an vertrauten starren Hierarchien beeinträchtigt wird. „Von der personalen Seite her können und müssen Fehler und Versäumnisse in der technischen und organisatorischen Gestaltung zwar immer wieder durch das '[...] Endgerät' Mensch und seine Flexibilität ausgeglichen werden, ein solches 'trouble

---

[1] vgl. erst **[Wild1973]** S. 45 dann **[Bleicher1990]**
[2] vgl. **[Bleicher1990]**, S. 55
[3] **[Krallmann1989]**, S. 1

shooting’ ist jedoch keineswegs motivierend“[4]. Vielmehr ist es sinnvoll, dem ausführenden Benutzer ein Werkzeug in die Hand zu geben, mit dem er flexibel vor Ort eine Fehlerquelle beseitigen kann. Dann muß ein erkannter Fehler oder eine bremsende Struktur nur einmal korrigiert werden, ohne daß sich dauerhafte informelle Verhaltensweisen herausbilden, die koordinierenden Instanzen meist verborgen bleiben.
Dynamische Umweltverhältnisse bedingen eine „[geringere] Stellenspezialisierung, weil inexakte Aufgaben und häufige Veränderungen breite Qualifikation und ganzheitliche Tätigkeiten erfordern. Sie fördern eine Tendenz zu ‘flüchtigen’ Formen der Aufgabenverteilung, z.B. der **Projektorganisation**“[5]. In diesem Rahmen ist eine möglichst flexible Gestaltung der Aufbauorganisation nötig, um die Ablauforganisation zu optimieren.[6]

Weiterführende Literatur zur Forderung der flexibleren Struktur der Unternehmens-Hierarchien bieten u.a. **[Aghte1989]**, **[Röthig1989]** und **[Staerkle1989]** unter dem Fachbegriff *Aufbauorganisation.*

Zielsetzung dieser Diplomarbeit ist es, eine bestehende Aufbauorganisations - Datenbank konzeptionell zu verfeinern und ein Software System zu entwickeln, mit dessen Hilfe die Aufbauorganisation graphisch bearbeitet werden kann. Sie soll damit eine Produktpalette ergänzen, in der eine Datenbankinfrastruktur und Werkzeuge zum Erstellen, Simulieren, Analysieren und Steuern von automatisch ablaufenden Geschäftprozessen, also der Ablauforganisation, angeboten werden.

Somit ist ein Werkzeug entstanden, das für alle beteiligten Mitarbeiter mit einem möglichst geringen Lernaufwand nutzbar sein soll. Die Aufbauorganisation wird mit einfachen graphischen Elementen visualisiert und die graphische Darstellung mit Funktionalitäten zur Benutzerinteraktion ausgestattet. Dahinter steht die Intention, das Unternehmen, besonders im Bereich der beteiligten und vernetzten Bildschirmarbeitsplätze, offen und transparent darzustellen. Jedem Mitarbeiter soll die Möglichkeit gegeben werden, für seinen Bereich eigenverantwortlich Strukturen aufzubauen oder Änderungen einzugeben. Damit ist eine bisher nicht gegebene Aktualität der Aufbauorganisation erreichbar. Die Ablauforganisation baut dann fortwährend auf eine optimal dokumentierte Aufbauorganisation auf, und diese kann anhand aktuell anfallender Geschäftsprozesse auch vor Ort flexibel reorganisiert werden.

Der inhaltliche Aufbau dieser Arbeit liefert mit **Kapitel 2** zunächst eine theoretische Grundlage, indem es wichtige Begriffe vorwiegend aus dem Bereich der Organisationstheorie definiert. Einige Theorieansätze der klassischen Organisationstheorie werden kurz vorgestellt und ein Bezug zum Stand der Büroinformationstechnik hergestellt. Nach dieser Vorbereitung beschreibt **Kapitel 3** zuerst das Umfeld, in das sich diese Arbeit einordnet. Danach folgt ein systematischer Aufbau der Konzepte, die zu den Strukturen des verwendeten Organisationsmodells geführt haben. Um das Modell zu

---

[4] **[Kötter1993]** S. 221 f.
[5] **[Picot1984]** S. 141
[6] vgl. **[Kötter1993]** S. 220 f.

vervollständigen werden darüber hinaus Möglichkeiten zur graphischen Präsentation erläutert.
Kapitel 4 behandelt die prototypische Umsetzung der Konzepte im *Organization-Object Modeler*. Dazu gehört zum einen die Darstellung der Datenbankstrukturen mit ihren vielfältigen Interaktionsmöglichkeiten, zum anderen folgt eine Beschreibung wichtiger Funktionalitäten der graphischen Benutzeroberfläche, der Kontrollstrukturen und der im Hintergrund ablaufenden Prozesse.
Den inhaltlichen Abschluß bildet **Kapitel 5** mit einer kritischen Würdigung und einem Ausblick auf weitere Entwicklungsmöglichkeiten der in dieser Arbeit vorgestellten Konzepte.
Die umfangreicheren Kapitel 3 und 4 bauen jeweils auf Konzepte der vorangegangenen Kapitel auf. Deshalb ist ihre genauere Aufgliederung an dieser Stelle schwierig. Sie beginnen jeweils mit einer Einführung und Erläuterung des inhaltlichen Aufbaus.

Der ausführliche **Anhang** enthält vorwiegend technische Informationen, die zum Verständnis der angewandten Methoden nicht unmittelbar erforderlich sind. Es werden Vorgehensweisen zum Installieren, Übersetzen oder zum Debugging der Applikation schrittweise erläutert. Die Strukturen der Quelldateien und implementierten Klassen werden aufgezeigt. Damit richtet sich der Anhang an Personen, die an der technischen Realisierung der Programmstrukturen interessiert sind. Im Anhang konnte jedoch keine erschöpfende Anleitung über Implementationsdetails gegeben werden, da eine genauere Beschreibung der fast 2,7 Megabyte großen Quelldateien den Rahmen dieser Arbeit gesprengt hätte. Ausführliche implementationstechnische Details stehen direkt in den Quelldateien.

Als prototypische Umsetzung der Konzepte ist im Rahmen dieser Arbeit die Applikation *Organization-Object Modeler* entwickelt worden. Sie ist in einer MS-Windows und einer OS/2 Version verfügbar. Es gibt für diese Betriebssysteme jeweils eine deutsche und eine englische Version. Eine spanische Version ist in technischen Details vorbereitet aber erst teilweise übersetzt. Während die theoretischen Grundlagen dieser Arbeit auf Forschungstätigkeiten an der Universität Paderborn aufbauen, sind Aspekte der Benutzerführung und Geschwindigkeitsanforderungen in Absprache mit der Pavone Informationssysteme GmbH, Paderborn, entstanden. Das Paderborner Softwarehaus möchte den Organization-Object Modeler in seine Workflow Produktpalette ‘GroupFlow’ aufnehmen.

Kapitel 2: Vorüberlegungen für Organisationsdesign mit flexiblen Strukturen

2.1 Theoretische Grundlagen

2.1.1 Begriffe aus der klassischen Organisationstheorie

Die klassische Organisationstheorie arbeitet mit einer eigenen Terminologie, deren Bedeutung z.T. von der umgangssprachlichen Verwendung abweicht. Auch gibt es unter den anerkannten Wissenschaftlern verschiedene Auslegungen und Intentionen über Sinn und Gehalt einiger der Fachbegriffe. Deshalb werden in diesem Kapitel einige Begriffe für die spätere Verwendung in ihrer Bedeutung festgelegt.

Auf jede Definition folgt zunächst eine klassische Definition, die sich meist an **[Picot1984]** oder **[Frese1976]** anlehnt, um einen Bezug zu Arbeiten der reinen Organisationstheorie aufzuzeigen. Einige dieser Definitionen werden danach für den Gebrauch im Umfeld der modernen Bürokommunikation abgewandelt. Für die weitere Diskussion innerhalb dieser Arbeit gilt dann die letztere Version, es sei denn, es wird ausdrücklich auf die klassische Theorie verwiesen.

2.1.1.1 Organisation

Es gibt grundsätzlich zwei verschiedene Sichtweisen, unter denen der Organisationsbegriff zu verstehen ist. Arnold Picot schlägt eine Unterteilung in einen instrumentellen und einen institutionellen Organisationsbegriff vor:

Der „[...] **instrumentelle[n] Organisationsbegriff:** Organisation als Inbegriff aller auf Aufgabenteilung und Koordination abzielender Regelungen ist Mittel zum Zweck der Zielerreichung der Unternehmung; zugleich ist sie ein Instrument zur Erreichung der individuellen Ziele derjenigen, die arbeitsteilig an der Erfüllung der Arbeitsaufgabe mitwirken.
Der **institutionelle Organisationsbegriff** hat einen ganz anderen Ausgangspunkt. Er knüpft an dem weitverbreiteten Phänomen der personellen Aufgabenteilung an: Die (Mehrpersonen-) Betriebswirtschaft als ein zielgerichtetes, bestimmten Regeln unterworfenes, soziales System ist eine Organisation; sie gleicht insofern anderen, nicht auf wirtschaftliche Ziele verpflichteten Gebilden (Verbände, Parteien, Kirchen usw.), die arbeitsteilige Ziele verfolgen[7].“

Während der institutionelle Organisationsbegriff („Die Unternehmung **ist** eine Organisation“) in der verhaltenswissenschaftlichen, deskriptiv ausgerichteten Organisationstheorie

---

[7] **[Picot 1984]** S. 98

vorherrscht, geht man in der Betriebswirtschaftslehre und in der Unternehmenspraxis durchweg von einem instrumentellen, gestaltungsorientierten Begriffsverständnis aus („Die Unternehmung **hat** eine Organisation“). Beide Anschauungen bedingen sich letztlich gegenseitig: Weil die Unternehmung eine Organisation im Sinne zweckgerichteter Regeln hat, ist sie eine Organisation im Sinne eines zielgerichteteten sozialen Systems.[8]

„Organisation kann ferner als Tätigkeit (**Organisieren**) oder als Ergebnis dieser Tätigkeit (**Regelsystem**) verstanden werden. Im ersten Fall geht es um den Prozeß der Bewältigung des oben beschriebenen Organisationsproblems, im zweiten um dessen Resultat.“[9]

Der Begriff der Organisation wird im folgenden in der instrumentellen und auch der institutionellen Bedeutung von Picot verwendet.

In der betriebswirtschaftlichen Organisationslehre des deutschen Sprachraums wird zumeist zwischen der **aufbau-** und der **ablauforganisatorischen** Perspektive unterschieden.[10] Aufbau und ablauforganisatorische Gestaltung unterscheiden sich hinsichtlich des Gestaltungsergebnisses. Während die Aufbauorganisation einer Unternehmung „die zu Abteilungen zusammengefaßten Stellen und die Beziehungen zwischen Stellen“[11] beschreibt, hat die ablauforganisatorische Sichtweise die räumlich-zeitlichen Zusammenhänge beim Vollzug von Aufgabenerfüllungsprozessen zum Gegenstand.[12]

„Bei der ablauforganisatorischen Gestaltung wird eine Aufgabe in einzelne Verrichtungen zerlegt, um unter Berücksichtigung der Aufgabenerfüllungssituation eine geeignete Reihenfolge zu bestimmen. Der Detaillierungsgrad ist damit höher als bei aufbauorganisatorischen Fragestellungen, bei denen die dauerhafte Regelung der Aufgabeninhalte einzelner Stellen, die Zusammenfassung der Stellen zu Subsystemen (Abteilungen, Unternehmensbereiche, Gruppen) und die Leitungs- und Kommunikationsbeziehungen im Mittelpunkt stehen“.[13]
Die Gestaltung der Aufbauorganisation [...] läßt sich als Entscheidungsproblem im Sinne der betriebswirtschaftlichen Entscheidungstheorie auffassen, bei dem organisatorische Merkmale, [...] so verändert werden, daß unter Beachtung von gegebenen, nicht beeinflußbaren Situationen ein möglichst hoher Zielerfüllungsgrad erreicht wird.“[14]

Eine Organisation läßt sich also
a) als eine Struktur zu einem Zeitpunkt darstellen,
b) oder es können sachliche, in Raum und Zeit ablaufende Prozesse beobachtet werden.[15]

---

[8] vgl. **[KiesKub1983]** S. 2, Zitate ebenda
[9] **[Picot 1984]** S. 98 f.
[10] vgl. **[Frese1976]**, **[Kosiol1976]** oder **[Grochla1975]** S. 24 f.
[11] **[Frese1976]**, S. 116
[12] vgl. **[Kosiol1976]** S. 22
[13] **[Dinkelbach1989]** S. 23, vgl. auch **[Frese1976]** S. 115 f., **[Grochla1975]** S. 24
[14] **[Dinkelbach1989]** S. 24
[15] vgl. **[Picot1984]** S. 99

Im Fall a) „stehen Teilaufgaben der Aufgabenträger und die zwischen diesen existierenden Beziehungen im Mittelpunkt. Man spricht dann von einer **Aufbauorganisation** (z.B. Art der gebildeten Aufgabenbereiche, Stellen, Abteilungen sowie den zwischen diesen bestehenden Weisungs- und Informationsbeziehungen).“[16]

Unter der Perspektive zeitlicher Abläufe, „die sich bei und zwischen den Aufgabenträgern vollziehen, [...] spricht man von **Ablauforganisation** (z.B. Stationen, Bearbeitungsvorgänge, Zeiten und Prioritäten bei der Abwicklung von Aufträgen).“[17] „Diese Unterscheidung betont zwei Seiten desselben Objektes: Ein organisatorischer Aufbau ohne Organisation der Abläufe ist sinnlos; organisatorische Abläufe ohne aufbauorganisatorischen Rahmen sind nicht vorstellbar.“[18]

Der Begriff der Aufbauorganisation wird im folgenden mit einer Einschränkung verwendet: Dinkelbach sprach 1989 noch von einer mehr dauerhaften Regelung der Aufgabeninhalte der Elemente der Aufbauorganisation (s.o.). Die Zielsetzung dieser Arbeit beinhaltet einen flexibleren Umgang mit der Aufbauorganisation bis hin zu flüchtigen Strukturen, so daß das Attribut ‘dauerhaft’ in Frage gestellt werden soll. Als Oberbegriff, der beide Organisationsbegriffe beschreibt, wird in dieser Arbeit der Begriff **Unternehmensorganisation** verwendet.

Unternehmensorganisation

Aufbauorganisation

Ablauforganisation

Abbildung 2-1: Zusammenhang von Organisationsbegriffen

[16] **[Picot1984]** S. 99
[17] ebenda, vgl. auch **[Kosiol1976]** S. 32 f.
[18] ebenda

2.1.1.2 Stellen

„Unter einer **Stelle** (oder auch Aktionseinheit) versteht man eine **personenunabhängige, abstrakt gedachte Einheit** von einem oder mehreren Aufgabenträgern, dem (denen) bestimmte Aufgaben-, Kompetenz- und Verantwortungsbereiche sowie die Verbindungswege zu anderen Stellen zugeordnet sind. Eine Stelle ist die **kleinste organisatorische Einheit** und muß durch mindestens eine Person als Aufgabenträger und Verantwortlicher besetzt sein [...]“.[19]

Die Definition ist in der Literatur nicht eindeutig, so beschränken andere Autoren die Größe einer Stelle auf die Kapazität einer Person, z.B. **[Kosiol1962]**. Im folgenden wird jedoch die Definition Schertlers berücksichtigt.

2.1.1.2.1 Stabstelle

Eine besondere Art von Stellen sind **Stabstellen**. Arnold Picot versteht darunter „Stellen, die im wesentlichen Ausführungs- und Verfügungskompetenzen für die Vorbereitung (Planung) und Überwachung (Kontrolle) von Entscheidungen haben, jedoch selbst keine Entscheidungs- oder Weisungsrechte besitzen [...]. Stabstellen sollen Instanzen bei der Informationsverarbeitung entlasten und durch Einbringung besonderen Sachverstands beraten.“[20]

Stabstellen sind demnach Stellen, die anderen Stellen zur Unterstützung zugeordnet werden. Sie stellen spezielles Wissen und Fähigkeiten zur Verfügung und dienen zur Kontrolle und Entscheidungsvorbereitung. Beispiele hierfür sind Qualitätszirkel oder juristische Einheiten. Da sie keine Weisungsbefugnisse besitzen, können ihnen auch keine Stellen direkt untergeordnet werden.

Erweiterungen des Stabstellenkonzeptes sehen vor, daß eine Stabstelle mehrere Stellen unterstützt. Auch wird vorgeschlagen, Stabstellenhierarchien aufzubauen, in der eine Rangordnung besteht. In dieser Arbeit wird eine Stabstelle immer nur einer Stelle angegliedert, und es ist keine Stabstellenhierarchie vorgesehen. Es ist jedoch möglich, daß eine Stelle mehrere Stabstellen besitzt.

---

[19] **[Schertler1988]** S. 26

[20] **[Picot1984]** S. 114

2.1.1.2.2 Stellengruppe

Ein Zusammenschluß mehrerer Stellen ist eine Stellengruppe, meist eine Abteilung oder eine Arbeitsgruppe:

„Mehrere Stellen - als kleinste organisatorische Einheiten - können zu sogenannten Stellengruppen bzw **Abteilungen** zusammengefaßt werden, die einer **Instanz** unterstellt sind (z.B. Abteilungsleiter) und die Erfüllung einer Gesamtaufgabe bzw. -funktion übertragen bekommen haben."[21] Zu dieser Definition fügt Erich Frese noch hinzu, daß Abteilungen potentiell unbefristet existieren sollen.[22] Abteilungen sind i.d.R. an einem oder wenigen Orten zentriert.

Abbildung 2-2 Abteilungen in einer Hierarchie

Arbeitsgruppen sind ein Zusammenschluß von Stellen, der sich über mehrere Abteilungen erstrecken kann, z.B. bei interdisziplinären Projektgruppen. Sie haben einen meist temporären Charakter und werden meist für spezielle Projekte gebildet. Die Arbeitsgruppenmitglieder sind meist Spezialisten, die sich in der Gruppe ergänzen. Deshalb sind Arbeitsgruppen flexibler als Abteilungen, da sie über breitgefächertes Wissen und Fähigkeiten verfügen. Die Mitglieder einer Arbeitsgruppe können geographisch an ganz verschiedenen Orten tätig sein.[23]

---

[21] **[Schertler1988]**, S. 26
[22] vgl. **[Frese1976]** S. 177
[23] vgl. **[Heinz1995]**, S. 20

Arbeitsgruppen können ineinander verschachtelt sein:

Abbildung 2-3: Verschachtelte Arbeitsgruppen[24]

2.1.1.3 Der Begriff der Hierarchie

Hierarchie gilt als universelles Strukturprinzip, das für eine Gesamtheit von Elementen systematische Beziehungen der Unter- und Überordnung schafft.[25] „In sozialen Systemen erzeugt Hierarchie über die bloße Funktionsteilung hinaus eine Differenzierung nach Rang, Status, Autorität, Befehlsgewalt, Entscheidungsbefugnissen u.a.m.“[26]

Oskar Grün sieht in der Hierarchie desweiteren eine „notwendige Voraussetzung für das zeitliche Überdauern sozialer Systeme.“[27] Die Hierarchie entsteht „[...] aus mindestens zwei Rängen, die in einem Über- oder Unterordnungsverhältnis stehen. [...] Die Rangordnung gliedert die Stellen unter dem Kriterium der Leitungsgewalt vertikal in einen Instanzenweg, der von der ranghöchsten bis zur rangniedrigsten Stelle läuft.“[28]

Bisher galten hierarchische Strukturen in Wirtschaftsunternehmen als effektivste Unternehmensformen. Das wird aber zunehmend in Frage gestellt, da schnelle Reaktionen auf Umwelteinflüsse u.a. durch Instanzenwege behindert werden.

2.1.1.4 Geschäftsprozesse

Ein Geschäftsprozeß beinhaltet die Gesamtheit und Aufeinanderfolge von Arbeitsschritten zum Erbringen einer Leistung für einen oder mehrere Kunden. Dabei kann der ‘Kunde’ eine andere Stelle innerhalb der Unternehmung sein. Ein Geschäftsprozeß kann aus mehreren kleineren Geschäftsprozessen bestehen, die in ihrer Gesamtheit der Zieler-

[24] Quelle: **[Heinz1995]**, S. 20
[25] vgl. **[Krüger1985]**
[26] **[Tannenbaumu.a.1974]**, S. 24 f.
[27] **[Grün1989]** S. 677
[28] ebenda

füllung des Unternehmens dienen, nämlich dem Erbringen einer marktgerechten Leistung. Meist wiederholt sich ein Geschäftsprozeß im Rahmen der Leistungserstellung eines Unternehmens.[29]

Zielgrößen für Geschäftsprozesse sind „Kundenorientierung, Durchlaufzeit, Reaktionsschnelligkeit und Flexibilität des Unternehmens“[30] Der dominierende Faktor ist somit *Zeit*. Daneben steht eine Verbesserung der Qualität oder der Kostensituation.

Im folgenden soll eine Spezialisierung des Begriffes ‘Geschäftsprozeß’ (engl. ‘Workflow’) auf den Bereich der Büroautomation verwendet werden. Wenn im Rahmen der Büroautomation von einem Geschäftsprozeß die Rede ist, versteht man darunter meist die Bearbeitung eines Dokumentes
- in verschiedenen Erscheinungsformen (verbal, animiert, textuell, graphisch o.ä.)
- durch verschiedene Stellen.

Ein Beispiel hierfür ist das Bearbeiten einer Einkommensteuererklärung im Finanzamt (in der Tat hinkt die Entwicklung der Büroautomation in diesem Bereich der öffentlichen Verwaltung der freien Wirtschaft bedenklich hinterher).

2.1.2 Aufgabe und Inhalt einer Organisationsdatenbank

„Eine Ansammlung von Daten, die in elementarer Beziehung zueinander stehen oder gemeinsame Merkmale haben, nennt man einen Datensatz. Ein Datensatz kann aus mehreren Datenfeldern bestehen.“[31] Betrachtet man den Informationsgehalt, so ist ein Datensatz „eine kleinste logisch in sich abgeschlossene Datenmenge“[32], z.B. Daten eines Mitarbeiters in einer Personalkartei:

Datensatz

Name | Abteilung | Wohnort | Tätigkeit

Datenfelder

Abbildung 2-4 Ein Datensatz

[29] vgl. **[HamCham93]**S. 3
[30] **[Fahrwinkel1995]**S. 3
[31] **[Duden1989]**S. 146
[32] ebenda

„Eine Folge von Datensätzen bildet eine **Datei**."[33] Eine Datei ist demnach eine Ansammlung von Datensätzen gleichen Typs. Mehrere Dateien bilden eine **Datenbank**. Eine Datenbank ist ein „System zur Beschreibung, Speicherung und Wiedergewinnung von umfangreichen Datenmengen."[34] Sie ist meist Teil eines größeren Informationssystems, das Daten von der Datenbank anfordert, sie be- oder verarbeitet und in veränderter Form zur Ablage zurücksendet. Größere Datenbanken besitzen neben den eigentlichen Daten meist ein Verwaltungssystem. Sie steuert als Schnittstelle den Zugriff anderer Applikationen auf die Daten und übernimmt Verwaltungsfunktionen.

Weil Daten nur einmal an einer Stelle abgelegt sind, bietet die Datenbank Vorteile in Datenredundanz, Schnelligkeit der Aktualisierung und Möglichkeit zu einheitlicher Kontrolle der Daten. Befinden sich Teile der Datenbank „auf verschiedenen Rechnern, wobei jeder Rechner über ein Rechnernetz auf den gesamten Datenbestand zugreifen kann, so spricht man von einer **verteilten Datenbank**."[35]

Die in dieser Arbeit betrachteten **Organisationsdatenbanken** beinhalten Daten, die die Aufbauorganisation eines Unternehmens beschreiben. Dazu gehören formale Stellen, abstrakte Rollen und auch deren Hierarchien und Beziehungen untereinander. So werden z.B. mit dem Datensatz einer Person auch deren Rolle und Stellung in Abteilungen und Arbeitsgruppen gespeichert. Da zwischen den Datenobjekten Beziehungen (Relationen) bestehen, sind die Organisationsdatenbanken sog. **relationale Datenbanken**.

2.1.3 Organisation und Informationstechnik

Im Bereich der Büroarbeitsplätze hat seit Ende der 70er Jahre ein tiefgreifender Wechsel in der Infrastruktur stattgefunden. Am Anfang der Büroautomation standen textorientierte Terminalgeräte ohne graphische Fähigkeiten (aufgrund ihrer grünen Bildschirme SUGs = Silly ugly greens genannt). Sie waren mit keiner eigenen Verarbeitungskapazität ausgestattet, sondern boten lediglich eine Verbindung zu großen Rechenmaschinen, den 'Mainframes'.

Mehrere Terminalgeräte mußten sich einen Mainframe teilen, der ihnen Rechenkapazität zur Verfügung stellte. Durch lange oder langsame Verbindungswege und aufgeteilte Rechenkapazität des Mainframes hatten diese Systeme meist lange Antwortzeiten. Dazu kamen nahezu kryptische Formulierungen, mit denen sich der menschliche Benutzer dem System mitteilen mußte.

Anders hingegen stellt sich der heutige Bereich der Bürokommunikation dar. Durch einen immensen Preisverfall der einzelnen Bauteile können Arbeitsplätze mit immer höherer Rechenkapazität ausgestattet werden. Die Fähigkeiten heutiger Rechenmaschinen in der optischen und akustischen Aufbereitung von Informationen sind derart gestiegen,

---

[33] **[Duden1993]** S. 166
[34] **[Duden1989]** S. 137 f.
[35] **[Duden1993]** S. 159

daß dem Benutzer eine multimediale, virtuelle ‘Modellwelt’ geboten werden kann. Dadurch haben sich die Möglichkeiten in der Gestaltung der Kommunikation zwischen Mensch und Maschine grundlegend gewandelt. Mußte sich der Mensch zu Zeiten der ‘green uglies’ auf das semantische Niveau der Maschine begeben, so ist es heute die Maschine, die sich dem Wahrnehmungsvermögen des Menschen anpassen muß, um eine ausreichende Akzeptanz zu erhalten und optimale Interaktion zu ermöglichen.

Exkurs:
Natürlich ist es nicht die Maschine, die sich Gedanken über das Wahrnehmungsvermögen des Menschen machen muß, sondern vielmehr die Entwickler der Applikationen. Ihre Tätigkeit wandelt sich immer weiter von einer mathematisch technischen Denkweise zu einer interdisziplinären Wissenschaft, die soziale, graphische, akustische und kognitive Kenntnisse einbringt, um den Forderungen der Anpassung der Maschine an den Menschen gerecht zu werden.
Dieser Perspektivenwechsel ist nicht nur, wie in der Einleitung provokativ beschrieben, durch Erfordernisse der wachsenden Produktivität des Einzelnen entstanden, sondern auch deshalb, weil der Mensch im Zuge der Humanisierung der Arbeitswelt nicht mehr als Produktionsfaktor angesehen wird und angesehen werden darf. Um seine Kreativität nutzen zu können muß er einen höheren Informationsstand haben. Es wird versucht, die Interaktionsschemata zwischen Mensch und Maschine auf ein Minimum zu begrenzen, damit der Mensch nicht durch erzwungene, sequentielle Vorgangsschritte in seiner Kreativität gebremst wird.
Anders formuliert: *Die Entwicklung der Informationstechnologie muß dahin wirken, daß der Mensch den Computer benutzen kann, ohne daß er erst aufwendig Kenntnisse für den interaktiven Umgang mit diesem Medium erwerben muß. Die Kreativität des Menschen soll sich nicht im* ***Umgang*** *mit der Maschine erschöpfen, sondern die eigentliche Intention, mit der er das Gerät anstellt, muß direkt vom Anfang der Interaktion an im Vordergrund stehen.*[36]

Die lokale Rechenkapazität jedes Arbeitsplatzes wird durch nahezu totale Vernetzung der einzelnen Rechenmaschinen erweitert. Sie ermöglicht Arten der Kommunikation, deren einfacheren Formen fast keinen zeitlichen und räumlichen Beschränkungen mehr unterliegen. So kann zum Beispiel ein Mitarbeiter einer Firma in Tel-Aviv (Israel) mit seinem Kollegen in Hursley (England) eine direkte Kommunikationsverbindung aufbauen, um Erfahrungen auszutauschen. Desweiteren ist es möglich, Datenbestände, also das Wissen einer Unternehmung, in verteilten Datenbanken zu halten. So können lokale Datenbestände einen schnellen Zugriff liefern und eine absolute Aktualität erreichen. Trotzdem steht dieses Wissen allen Mitarbeitern auf Anfrage und bei entsprechender Zugriffsberechtigung zur Verfügung.

Aus diesen Möglichkeiten der Informationstechnologie entstehen Anforderungen an moderne Bürokommunikations- und Büroinformationskonzepte. „Dazu gehört insbesondere eine integrierte, von einzelnen Funktionsbereichen möglichst unabhängige Informa-

[36] vgl. z.B. **[Keil-Slawik1990] [Keil-Slawik1992] [WinRiehm1985]**

tionsverarbeitung und Datenhaltung. Die Struktur der Datenorganisation sollte objektorientiert und bereichsübergreifend angelegt sein und verschiedene Sichtweisen berücksichtigen.[37]

Bei aller Euphorie sollen aber einige hemmende Faktoren nicht unterschlagen werden. Der Bereich computerunterstützter kooperativer Arbeitsprozesse (engl.: Computer Supported Cooperative Work [**CSCW**]) hat erst seit Beginn der 90er Jahre von einer akademischen Disziplin den Einzug in die unternehmerische Praxis gefunden.[38] Insbesondere große Versicherungsunternehmen bildeten die Vorreiter dieser Entwicklung. Sie waren als erste vom praktischen Nutzen der „integrierten Vorgangsbearbeitung", dem „Workflow-Processing", um nur einige der Modewörter aus diesem Bereich zu nennen, zu überzeugen (Versicherungsunternehmen sind in der ersten Linie gekennzeichnet durch hohe Frequenz an immer wiederkehrenden Geschäftsprozessen und einer großen Datenmenge mit sehr strukturierten Datensätzen. Sie sehen sich mit einer immensen Papierflut konfrontiert, die noch Ende der 80er Jahre in bis zu fußballplatzgroßen, feuergefährlichen Archiven gespeichert wurde.[39]
Weiterhin ist der Aufbau von Infrastrukturen in betrieblichen Informations- und Kommunikationssystemen, insbesondere im Bürobereich, von recht einer recht konservativen Kostenrechnung geprägt. Dabei muß eine Infrastruktur in möglichst kurzer Zeit mit möglichst geringen Mitteln eine spürbare Ertragsteigerung bewirken. Ferner ist zu beachten, daß Computer zwar preiswerter, nicht jedoch billiger werden. Es wird von ihnen eine immer größere Leistungsfähigkeit erwartet, z.B. im multimedialen Bereich. Gesunkene Preise von Einzelbauteilen werden durch die Verwendung zusätzlicher oder leistungsfähigerer Komponenten kompensiert.

Anschaffungen im Bürobereich verursachen primär Gemeinkosten, die verkauften Stückzahlen oder Gewinnen nicht direkt zugeordnet werden können. So muß sich in vielen Unternehmen die Hard- und Softwareausstattung nach althergebrachten Abschreibungsregeln erst amortisieren, bevor neuere Technologie eingekauft wird. Auch geschieht die Aufrüstung dann nicht in einem Schritt, sondern es muß ein Nebeneinander vieler Technologien und Plattformen berücksichtigt werden.

Folglich kann nicht in allen Bereichen der Wirtschaft und in allen Unternehmen auf eine optimale Hardware aufgebaut werden. Die Entwicklung im Bereich der CSCW, die vorwiegend auf sehr leistungsfähige Software aufbaut, muß den infrastrukturellen Rahmen berücksichtigen, in dem sie später eingesetzt wird.

Was bedeutet das für diese Arbeit?

Es ist fortwährend zu berücksichtigen, daß die Repräsentation graphischer Strukturen und Interaktionsvorgänge auch auf marktüblichen Hardware - Plattformen der unteren Leistungsklasse optimal funktionieren. Wahrscheinlich beansprucht die graphische Dar

---

[37] **[Fischer u.a.1994]** S. 199
[38] vgl. ebenda S. 273
[39] vgl. **[PlesBart1990]**

stellung am meisten Rechenkapazität. Daher muß ein Weg gefunden werden, mit nicht zu aufwendigen graphischen Objekten eine leistungsfähige, virtuelle Modellwelt zu schaffen, die mit geringen Antwortzeiten bearbeitet werden kann.

2.2 Praktische Grundlagen, worauf baut diese Arbeit auf?

Wie im vorhergehenden Abschnitt beschrieben, besteht die Infrastruktur vieler Unternehmen in einem Nebeneinander vieler Architekturen und Techniken. Die Hardware setzt sich aus verschiedenen Netzwerktechniken und Vernetzungstopologien sowie verschiedenen Rechnerarchitekturen mit unterschiedlicher Leistungsfähigkeit zusammen. In der Software gibt es verschiedene Betriebssysteme und Applikationen, die zum Teil recht unterschiedliche Standards entwickelt haben.

Die im Bereich der computerunterstützten kooperativen Arbeitsprozesse (CSCW) angebotene Software, die sog. 'Groupware', muß also auf eine Vielzahl von Hardwareplattformen aufbauen. Einer der Vorreiter ist die Firma Lotus. Ihre Notes-Groupware wird auf PCs mit den Microsoft und IBM Betriebssystemen, Macintoshrechnern und UNIX-Systemen verwendet. Als Datenbankserver können zusätzlich Novell Netware Server benutzt werden. Mit diesen Systemen ist ein großer Bereich der heutigen Infrastruktur in Unternehmen abgedeckt. Dank innovativer offener Datenhaltung in verteilten Datenbanken, die mit eigener Informationsverarbeitungskapazität ausgestattet sind, konnte sich Lotus als anerkannter Marktführer etablieren.

Die Paderborner *PAVONE Informations System GmbH* baut auf die Lotus Groupware auf und versucht, die Vorteile der Groupware mit dem heutigen Stand der Büroautomatisationssysteme zu verbinden. Daraus ist die Produktpalette *GroupFlow* entstanden. Die Architektur und Konzepte von GroupFlow werden eingehend in **[NastHilp1994a]** und **[NastHilp1994b]** beschrieben. Deshalb soll an dieser Stelle nur eine minimale Einführung stehen.

Mit den Produkten der GroupFlow Palette können Büroabläufe modelliert, simuliert, automatisch gesteuert und analysiert werden. Die Vorgangsbearbeitung erfolgt dokumentgestützt, d.h. Daten werden als Dokumente in verschiedener Darstellung bearbeitet und weitergeleitet. Innerhalb eines objektorientierten Geschäftsprozesses, dem Workflow, werden einzelne Bearbeitungsschritte anhand dieser Dokumente durchgeführt oder es werden separate Softwareprodukte aufgerufen.[40]

Die zur Bearbeitung nötigen organisatorischen Einheiten, in unserer Terminologie 'Stellen', werden in einer Aufbauorganisationsdatenbank definiert. Die Entwickler von Workflows und auch real ablaufende Workflows greifen auf Informationen dieser Datenbank zurück. Das gesamte Wissen über strukturellen Aufbau des Unternehmens und

[40] vgl. z.B. **[Meyer1995]** Kap. 2 oder **[Heinz1995]**

deren Stellen wird in der Aufbauorganisationsdatenbank abgelegt. Es hat sich gezeigt, daß Workflows selten auf einzelne Abteilungen beschränkt ablaufen. Deshalb gibt es bisher *genau eine* Organisationsdatenbank, damit jedem Benutzer Informationen über die gesamte Aufbauorganisation ständig zur Verfügung stehen.

Bisher wird die Aufbauorganisationsdatenbank direkt unter Lotus Notes bearbeitet. Extremer Nachteil bei administrativen Tätigkeiten ist dabei das Fehlen graphischer Transparenz. Es können nur einzelne Datensätze über Makrobefehle aufbereitet werden, ohne daß durch interaktive Rückkopplung die Auswirkungen auf die organisatorische Struktur sichtbar sind. In einigen Bereichen ist die Bearbeitung der Datensätze in der Benutzerführung umständlich und langsam.

Zur Bewältigung dieses Mißstandes müssen im Organisationsmodell die zur Bewältigung der administrativen Tätigkeiten notwendigen Elemente identifiziert, beschrieben und in Zusammenhang gebracht werden. Darauf baut die Entwicklung eines objektorientierten Metamodelles auf, dessen Konzepte in einem graphischen Modellierwerkzeug realisiert wurden. Damit fungiert die Aufbauorganisationsdatenbank nicht nur als Datenquelle für die Beschreibung organisatorischer Einheiten. Durch visuelle, ‘erfahrbare’ Darstellung soll es außerdem möglich sein, die reale Aufbauorganisation zur Steigerung der Effektivität zu reorganisieren.

Zum Abschluß dieses Kapitels soll nach der theoretischen Vorarbeit noch ein bildlicher Eindruck vermittelt werden. In nachfolgendem Bildschirmausschnitt ist ein Teil eines Groupware Systems zu sehen (in diesem Fall GroupFlow von PAVONE). Es arbeitet auf drei Datenbanken: einer Aufbauorganisationsdatenbank, einer Ablauforganisationsdatenbank mit abstrakten und einer Anwendungsdatenbank mit real ablaufenden Arbeitsabläufen.

Während die Ablauforganisation mit graphischer Interaktion bearbeitet werden kann (Applikation im Fenster rechts unten, Modellierung einer Kfz-Schadensregulierung), sind die organisatorischen Einheiten, wie hier die Arbeitsgruppe ‘Telefonische Datenerfassung’ nur über Eingabemasken innerhalb der Datenbank zu bearbeiten (Fenster links). Der Entwickler nutzt die Möglichkeiten organisatorischer Strukturen nicht aus, da er kaum Informationen über Zusammenhänge und Hierarchien bekommt.

Abbildung 2-5 Bisheriges Zusammenspiel der Ablauforganisation mit der Aufbauorganisationsdatenbank

Kapitel 3: Organisationsmodell zur Gestaltung einer flexiblen Aufbauorganisation

Innerhalb dieses Kapitels soll die Vorgehensweise nachvollzogen werden, die zur Konzeption eines objektorientierten, also relationalen Metamodells geführt haben, mit dem eine möglichst große Anzahl von realen Organisationsformen dargestellt werden kann.

Im ersten Teil geht es darum, die wesentlichen Strukturmerkmale herauszuarbeiten. Dabei wird die Frage aufgeworfen, ob ein sinnvolles Nebeneinander von Hierarchie oder Gruppendynamik erzeugt werden kann. Projektorganisation in Arbeitsgruppen kommt der Idealvorstellung eines flexiblen Systems sehr nahe, aber ohne Hierarchie ist eine leistungsgerechte Fokussierung auf die ökonomischen Unternehmensziele scheinbar nicht gewährleistet.
Als Ergebnis der zuvor klassifizierten Anforderungen und Limitationen wird im zweiten Abschnitt das Organisationsmodell vorgestellt. Es werden zunächst wesentliche Objekte charakterisiert und jeweils in einem Entity - Relationship - Modell klassifiziert. Der dritte Teil dieses Kapitels widmet sich den Grenzen und Möglichkeiten graphischer Darstellung am Bildschirm. Es zeigt sich, daß nicht alle wünschenswerten Formen realisierbar sind, da Extremfälle nicht darstellbar werden oder die Rechenkapazität des Computers überfordern.

Daraufhin stellt das Kapitel 3.4 die graphische Präsentation vor, mit der die organisatorischen Elemente auch am Bildschirm dargestellt werden können. Dieser Teil zeigt dabei schon Teile der graphischen Implementation, aber noch keine Interaktionsmöglichkeiten, die **Kapitel 4** vorbehalten bleiben.[41]

Als Abschluß dieses Kapitels folgt ein Exkurs über mögliche Auswirkungen und künftige Möglichkeiten im Bereich der sozialen Arbeitsgestaltung, die sich mit dem Einsatz der CSCW Techniken und insbesondere durch eine partizipative Organisationsgestaltung ergeben können.

---

[41] vgl. Kap. 4.4.2 und 4.5

3.1 Anforderungen an Struktur und Inhalt einer Aufbauorganisations datenbank

3.1.1 Theoretische Implikationen

3.1.1.1 Ein Modell als Abstraktion der Realität

Wie im vorhergehenden 2. Kapitel beschrieben, ist eine Abbildung existierender Organisationsstrukturen eine Abstraktion der Realität. Solch ein virtuelles Modell zeigt reale Zusammenhänge in einer begrenzten Anzahl von Perspektiven. Es bleiben Details und Aspekte unberücksichtigt, die mit der Zielsetzung des Modells scheinbar nichts zu tun haben. Das ist in vielen Fällen angebracht, da die meist unerschöpfliche Masse an Details wesentliche Informationen überdecken und als unwichtig erscheinen lassen kann. An dieser Stelle wird untersucht, was ein Modell einer Aufbauorganisation können sollte und ob es verläßliche Richtlinien zur Abstraktion geben kann.

Eine geeignete Fokussierung auf organisatorisch relevante Aspekte ist mit Sicherheit sinnvoll. Sie bestimmt die Effizienz der Arbeit mit dem Modell und zeigt dem späteren Benutzer, welche Auswahlentscheidungen ihm zur Verfügung stehen. Andererseits werden in einem Modell Strukturen festgelegt, die sich in einer veränderten Umweltsituation als ineffektiv erweisen können. Aber wo ist die Grenze zwischen sinnvoller Abstraktion und einer losgelösten Darstellung, die den Zusammenhang zwischen Modell und Wirklichkeit nicht mehr zeigt?

In der Tat gibt es keine exakte Grenze. Das Einfühlungsvermögen und der Wiedererkennungseffekt ist von Person zu Person unterschiedlich ausgeprägt. Die Perspektive eines Menschen richtet sich nach seinem Blickwinkel beim Erleben und Handeln, beeinflußt durch seinen persönlichen *Bezugsrahmen*, der sich aus Erlebnissen, Erfahrungen und Ausbildung entwickelt.[42] Auf der anderen Seite orientiert sich eine realistische Struktur u.U. zu sehr an der Wirklichkeit und beschneidet damit die Möglichkeit zu kreativer individueller Gestaltung.

Arne Raeithel kommt in diesem Zusammenhang zu dem Schluß, daß es „keine Richtlinien für eine bestimmte Auswahl der verfügbaren operativen Mittel und einem einschränkenden Muster des Zusammenwirkens dieser Mittel [gibt], das dennoch in diesen Grenzen die nötige Flexibilität bei ihrer Nutzung garantiert.“[43]

Aus diesen Ausführungen kann man folgendes Fazit ableiten:
Jedes Informationssystem und damit jedes organisatorische Modell, „mag es auch noch so weit ausgebaut sein, ja die Welt umspannen, ist immer noch eine radikale *Strukturre-*

[42] vgl. **[Pasch1991]** S. 114
[43] **[Raeithel1991]** S. 136

duktion und *-amplifikation* gegenüber der repräsentierenden Wirklichkeit."[44] Ein organisatorisches Modell muß prinzipiell in der Lage sein, jedem Benutzer einen Realitätsbezug zu vermitteln. Gleichzeitig darf es keinen zu großen Einfluß auf den Umgang mit dem System nehmen. Allein schon aus Gründen der Verarbeitungskapazität der Rechenmaschinen kann ein Organisationsmodell keine detaillierte Abbildung aller realen oder möglichen Strukturen liefern.

Bedeutsam für das Organisationsmodell ist demnach nicht, *wie* eine organisatorische Einheit dargestellt wird. Dafür ist der Wiedererkennungseffekt anhand rein optischer Kriterien zu gering. Statt dessen sollte ein sozialorientiertes Modell die wesentlichen Zusammenhänge *erfahrbar* machen. D.h. der Schwerpunkt der Modellbildung darf nicht in der rein graphischen Ausgestaltung eines oder mehrerer organisatorischer Objekte liegen. Vielmehr ist es wichtig, die einzelnen Objekte mit einem interaktiven Regelsystem auszustatten, das die formalen Konsequenzen einer Handlung in einer Weise zeigt, wie sie auch in der realen Organisation stattfinden würden.

Desweiteren ist eine Navigationskomponente hilfreich, die zwischen globalen Perspektiven und lokalen, detaillierteren Darstellungen wechseln kann. In einer gröberen Darstellung sind insgesamt mehr organisatorische Zusammenhänge zu sehen. Dadurch kann der Benutzer globale Einstiegspunkte finden, um dann spezielle Teilstücke zu vergrößern und zu bearbeiten. Optimal wäre eine stufenlose Möglichkeit der Vergrößerung oder Verkleinerung (engl.: Zooming).

Letzte Forderung ist eine Möglichkeit zur hierarchischen Modularisierung. Das unterstützt die bei fast jeder Entwurfstätigkeit üblichen **Bottom-Up** und **Top-Down** Vorgehensweisen. Erstere definiert zunächst Details und aggregiert diese dann zu Metastrukturen. Top-Down geht den umgekehrten Weg und definiert Aufgabenkomplexe, die in kleinere Unterschritte unterteilt werden. Obwohl die Top-Down Methode allgemein favorisiert wird und als zuverlässiger gilt, werden in der Praxis meist beide Methoden gemischt angewendet, indem man eine Struktur, sowohl vom Problem her als auch mit Blick auf die infrastrukturellen Möglichkeiten, aufbaut.[45] Bekannte Strukturen können so sofort in allen Einzelheiten ausformuliert werden, während neuartige oder komplexe Konstrukte im systematischeren Top-Down Ansatz erst erarbeitet werden. Daraus ergibt sich die Forderung nach einer mächtigen Aggregationsmethode, mit der einzelne Unterstrukturen ausgeblendet werden können.

Das Nebeneinander von Top-Down und Bottom-Up soll das folgende Beispiel verdeutlichen. Während die Unterstruktur von Abteilung A gerade bearbeitet wird, existiert bereits eine Unterstruktur der Abteilung B (durch das Pluszeichen erkennbar), die ausgeblendet wurde (Top-Down Ansatz, da zuerst Oberabteilung A vorhanden war). Es ist noch keine gemeinsame Oberabteilung von A und B definiert. Das entspricht der Bottom-Up Methode.

---

[44] **[Steinmüller1993]** S. 221
[45] vgl. **[Duden1989]** S. 550 f.

Abbildung 3-1: Hierarchische Modularisierung

3.1.1.2 Abwägung zwischen ‘mechanistischer’ und ‘organischer’ Organisation

In der klassischen Literatur zur Organisationstheorie wird zwischen *mechanistischer* und *organischer* Organisation unterschieden. In Organisationen mit homogener, stabiler Umwelt wird die mechanistische = hierarchisch strukturierte Organisation favorisiert.[46] Die organische Struktur scheint im Umfeld dynamischer Umwelteinflüsse effizienter zu sein.[47]

Die Zielsetzung dieser Arbeit ist, eine flexible Struktur für sehr dynamische Umwelteinflüsse zu konzipieren. Die Umwelteinflüsse ergeben sich im Umgang mit den Geschäftsvorfällen, beeinflussen also primär die Ablauforganisation. Eine flexibel auf Umwelteinflüsse reagierende Aufbauorganisation hat sich damit den Erfordernissen der Ablauforganisation anzupassen, muß sich ihr evtl. sogar unterordnen. „Da die Einheiten der Aufbauorganisation Ressourcen bzw. Prozeßträger sind, haben die **Struktur der Aufbauorganisation** und die aufbauorganisatorische Verankerung der Geschäftsprozesse maßgeblichen Einfluß auf die Prozeßeffizienz.“[48]

Wie die Anpassung der Aufbauorganisation an die Geschäftsprozesse der Ablauforganisation erfolgen soll, ist nicht eindeutig beantwortet. Für eine umfassende Optimierung von Geschäftsprozessen muß die Aufbauorganisation der Ablaufstruktur optimale Ressourcen zur Verfügung stellen können. Anstelle rein funktionsorientierter Betrachtungsweise könnte man dazu übergehen, die bisher relativ fixen Vorgaben der Aufbauorganisation anhand konkreter Geschäftsprozesse zu gestalten. Dadurch wird die Aufbauorganisation abhängig von den Gegebenheiten und Erfordernissen der

[46] vgl. **[Staehle1991]** S. 434
[47] vgl. **[WainFran1984]** S. 33
[48] **[Funk1992]** S. 357

Ablauforganisation. Es liegt dann nahe, hierarchische Verantwortung mit der Verantwortung für Prozeßstrukturen zu verknüpfen.

Solch eine Organisationsform beschreibt eine mehr temporäre Struktur, die sich für das Bearbeiten von Prozessen bildet und sich, je nach Umweltdynamik, schnell ändern kann. Das entspricht den in Kapitel 2.1.1.3 definierten Merkmalen von Arbeitsgruppen. Somit ist unter den gewählten Zielvorgaben (hohe Umweltdynamik mit temporären Aufgabenstrukturen) eine **Projektorganisation** in Arbeitsgruppen effizienter. Projektorganisation sollte jedoch keineswegs als Allheilmittel angesehen werden. Ihre enthierarchisierende Wirkung und die damit verbundene Kreativitätssteigerung haben ihre Grenzen. Von einer Beseitigung der Hierarchie kann auch in Gruppenstrukturen keine Rede sein, bestenfalls von einer Reduzierung des Machtgefälles. Die Praxis zeigt immer wieder, daß sich ein Teamleiter als notwendig erweist oder durch Gruppenprozesse informell herausbildet.[49]

Es ist aber die Frage, ob eine organisatorische Struktur nur aus Projektteams gebildet werden kann. Sicherlich ist diese Form nur bei kleineren Organisationsformen möglich. Bei großen Unternehmen beruht die Leistungserstellung auf Motivationsformen der Hierarchie, die gewisse Kontrollstrukturen und Aufstiegsmechanismen voraussetzen. Sie besitzen einen Verwaltungsapparat mit immer wiederkehrenden, also iterativen, administrativen Tätigkeiten. Dazu gehören u.a. Abteilungen zur Buchhaltung und zum Personalwesen. Zwar ist eine gewisse Kreativität z.B. in der steuerrechtlichen Buchhaltung durchaus sinnvoll, aber wohl eher die Ausnahme. Auf eine hierarchische Struktur können große Unternehmen demnach nicht verzichten.

Dennoch haben insbesondere größere Unternehmen die finanziellen Mittel für eine aufwendige automatisierte Büroorganisation. Für kleinere produzierende Unternehmen ist die Effizienzsteigerung durch Büroautomation i.d.R. zu gering, um hohe Investitionen in CSCW Technologie zu rechtfertigen. So sind es vor allem kleinere Unternehmen in Dienstleistungsbereichen und große Unternehmen, die als Abnehmer in Frage kommen.

3.1.2 Einschränkungen durch das Umfeld der PAVONE Groupware

Dem Organisationsmodell sind eine Reihe technischer Grenzen und Rahmenbedingungen gesetzt. Die prototypische Umsetzung, der ‘Organization-Object Modeler’, soll im Umfeld einer bestehenden Produktpalette zum Einsatz kommen.[50] Die Applikation ‘Organization-Object Modeler’ erzeugt und verwaltet eine Aufbauorganisation, auf die die anderen Produkte zugreifen. Deshalb sollte er nur Organisationsformen erzeugen, die die anderen Applikationen weiterverwenden können.

---

[49] vgl. **[Wild1973]** S. 52
[50] vgl. Kap. 2.2

Aus diesem Zusammenspiel ergeben sich vorrangig zwei Prämissen, von denen das zweite die Wahl der Organisationsform beeinträchtigt:

1. **Eine Verschachtelung von Arbeitsgruppen darf nur über höchstens 5 Ebenen verlaufen.**
   Die GroupFlow Palette arbeitet auf Grundlage des Notes Systems der Firma *Pavone*. Notes arbeitet mit Adreßbuch - Datenbanken, in denen Personen und Arbeitsgruppen abgelegt werden. Diese Adreßbücher erlauben nur eine Schachtelung von Arbeitsgruppen über 5 Ebenen, d.h. eine Arbeitsgruppe kann andere Unterarbeitsguppen enthalten, aber die gesamte Hierarchie darf nur 5 Ebenen umfassen. Dieser Limitation hat sich die GroupFlow Palette angeschlossen.

   Diese Beschränkung ist nicht schwerwiegend, da eine zu starke Schachtelung von Arbeitsgruppen dem Wesen der Projektorganisation entgegenwirkt. Eine zu ausgeprägte formale Struktur behindert die Flexibilität und Kommunikation der Beteiligten. Sollte es sich als nötig erweisen, kann die Kompatibilität zu den Notes - Adreßbüchern aus der Sicht des Organisationsmodells einfach aufgegeben werden.

2. **Einordnung einer Abteilung nur durch den Namen ihrer vorgesetzten Abteilung**
   Schwerwiegendere Konsequenzen hat eine andere Einschränkung durch das GroupFlow System: Abteilungen werden nur anhand einer Pfadvariablen identifiziert, d.h. eine Abteilung kann nur eine vorgesetzte Abteilung haben. Dadurch wird eine Organisationsform gewählt, die augenscheinlich nicht die flexibelste Form von Abteilungshierarchie ist. Diesem Aspekt ist das folgende Kapitel gewidmet, in dem das sog. **Einliniensystem** mit alternativen Organisationsformen verglichen wird.

   Die alternative Hierarchieform zu Einliniensystemen sind sog. **Mehrliniensysteme**, in denen eine Abteilung mehrere vorgesetzte Stellen haben kann. Bestehende GroupFlow - Produkte könnten wahrscheinlich mit geringem Aufwand abgeändert werden, indem jede Abteilung eine Liste ihrer direkt vorgesetzten Stellen bekäme. Eine Erweiterung des ‘Organization-Object Modelers’ auf Mehrliniensysteme erscheint aufwendig, da alternative Bildschirmdarstellungen entwickelt werden müßten (vgl. Kap. 3.3). Eine detailliertere Abwägung zwischen sog. Ein- und Mehrliniensystemen folgt im nächsten Kapitel.

3.1.3 Hierarchische Organisationsformen

Die klassische Diskussion über Organisationsformen wird von Verfechtern der sog. Einlinien- und der Mehrliniensysteme geführt. Sie ist durch zwei frühe Organisationstheoretiker, F.W. Taylor und Henri Fayol geprägt. Taylor favorisiert die Mehrlinienorganisation und während Fayol dagegen die Einlinienorganisation bevorzugte. Über diesen Bereich der Organisationslehre sind vielfältige Abhandlungen geschrieben worden.[51] Die Frage,

[51] vgl. z.B. **[Frese1976]** Kap. 822.3 und 92.3, **[Staehle1991]** S. 666

welches dieser Systeme anzuwenden ist, bleibt ein offenes Problem. Es scheint, als wenn Mehrliniensysteme für einige Fälle effizientere Strukturen bieten können. Im Rahmen dieser Arbeit soll jedoch ein möglichst universelles System entwickelt werden, für das sich eher Einliniensysteme empfehlen. Deshalb wird im folgenden vorrangig untersucht, ob ein Verzicht auf Mehrliniensysteme eine große Einschränkung der organisatorischen Möglichkeiten mit sich bringt.

Unter einem Liniensystem versteht man die hierarchische, formale Verknüpfung von Abteilungen. Es besteht eine bestimmte Rangordnung, die in der graphischen Darstellung durch Linien angegeben wird und meist von oben nach unten verläuft. Linien in dieser Richtung bezeichnet man als **Befehlslinien**. Die gleichen Linien beschreiben in umgekehrter Richtung den sog. **Dienstweg**, der für Meldungen, Mitteilungen und Beschwerden benutzt wird. Verschiedene Ausprägungsformen der Liniensysteme unterscheiden sich durch die spezifische Art der Verknüpfung zwischen den Stellen.

3.1.3.1 Einliniensysteme

Im Einliniensystem ist eine Stelle nur einer übergeordneten Stelle verantwortlich und bekommt ausschließlich von ihr direkte Weisungen.[52]

Abteilung A

Abteilung B

Abbildung 3-2 Zwei Abteilungen im Einliniensystem

Im Einliniensystem gilt das sog. Prinzip der „**Einheit der Auftragserteilung**“[53]. Damit ist gemeint, daß eine Stelle von höchstens einer übergeordneten Stelle Weisung bekommt. Für das Gelingen dieser Aufträge ist damit immer *nur eine* bestimmte Stelle verantwortlich. Die formellen Kommunikationswege verlaufen vertikal, mit Ausnahme einer in Sonderfällen vorgesehenen horizontalen Verbindung, einer sog. *Fayolschen Brücke*.[54]

[52] vgl. **[Frese1976]** S. 173 und Abbildung 3-2
[53] **[Staehle1991]** S. 660
[54] vgl. **[Staehle1991]** S. 660

Abbildung 3-3 Eine Einlinienhierarchie

3.1.3.2 Mehrliniensysteme

Im Mehrliniensystem kann eine Stelle mehreren ranghöheren Stellen untergeordnet sein.[55]

Abbildung 3-4: Abteilung mit mehreren übergeordneten Stellen

Mehrliniensysteme verzichten auf die Einheit der Auftragserteilung zugunsten einer funktionellen Unterteilung. Die vorgesetzten Stellen bekommen meist nur für bestimmte Funktionen Weisungsbefugnisse. Wenn eine Aufgabe in mehrere Funktionsbereiche fällt, erhält eine untergeordnete Stelle von mehreren vorgesetzten Stellen Anweisungen.[56]

Besondere Ausprägungen der Mehrliniensysteme sind **Matrix-** und **Tensororganisationen**. Bei ersterer wird „simultan zwei Aspekten eines Problembereiches Aufmerksamkeit geschenkt [...] (z.B. Verrichtung und Objekt) und bei der Tensor-Organisation sogar drei Aspekten (Verrichtung, Objekt, Region)“.[57] Beide Formen versuchen, das Prinzip der Hierarchie zu umgehen, um kürzere und kompetentere Kommunikationswege zu erhalten.

---

[55] vgl. Abbildung 3-4
[56] vgl. **[Staehle1991]** S. 661
[57] ebenda

Beispiel:
In Abbildung 3-5 ist eine Matrixorganisation abgebildet. Die Spalten sind nach Verrichtung, die Zeilen nach dem Objekt aufgeteilt. Die unten stehende mehrfarbige Abteilung „Produktion Spülmaschinen“ bekommt Weisungen zur Produktion von der oben stehenden Produktionsabteilung und von einer links stehenden Abteilung „Spülmaschinen“.

Abbildung 3-5: Eine Matrixorganisation[58]

Arnold Picot behauptet, daß besonders die Matrixorganisation in der Praxis eine Verbreitung gefunden hat.[59] Dagegen zeigt Wolfgang Staehle anhand empirischer Untersuchungen, „daß zumindest in Deutschland zwischen dem Anspruch von Unternehmungen, über eine Matrix-Organisation zu verfügen, und dessen realer Ausformung eine große Diskrepanz besteht [bestünde].“[60]

3.1.3.3 Stab - Liniensysteme

Zu der Einlinienorganisation existiert noch eine Erweiterung, die spezialisierte Stabstellen[61] zur Unterstützung leitender Stellen einsetzt. Das daraus resultierende **Stab-Linien-Modell** wird in der Literatur als Ansatz zur Verbindung der Ein- und Mehrlinienorganisation diskutiert.[62] Der Vorteil liegt in einer Entlastung der leitenden Stellen von Routinearbeiten und in einem fachlich fundierteren Wissensstand der Entscheidungsträger.

[58] Quelle: in Anlehnung an **[Brings1976]** S. 73
[59] vgl. **[Picot1984]** S. 120
[60] **[Staehle1991]** S. 666
[61] vgl. Kap. 2.1.1.2.1
[62] vgl. **[Staehle1991]** S. 660, **[Picot1984]** S. 118

Es entstehen aber auch Nachteile durch die Verwendung von Stäben. Zum einen können Stäbe möglicherweise einen Informations- oder Qualifikationsvorsprung realisieren und dadurch zu großen Einfluß ausüben. Zum anderen arbeiten Stäbe „in manchen Fällen wegen Kooperationsschwierigkeiten mit der [Leitungsstelle] praktisch ‘für den Papierkorb’ und fühlen sich wegen fehlender Entscheidungsbefugnis bei zugleich hoher Qualifikation frustriert.“[63]

Abbildung 3-6 Stab-Linien Modell der Organisationsform

3.1.3.4 Beurteilung der Organisationsformen

Vorteile von Einliniensystemen ergeben sich durch direkte Zuordnung von Weisungsbefugnissen. Eine Stelle ist für die ihr untergeordneten Stellen direkt verantwortlich. Es kann eine direkte Zuordnung bei Fehlfunktionen stattfinden, und die Sorgfalt bei der Erteilung von Weisungen wird besser sein. Daraus resultiert vordergründige Konfliktfreiheit, da immer eindeutig ist, welche Stelle zur Schlichtung von Konflikten herangezogen werden kann. „Auftretende Konflikte werden bis zu der hierarchischen Ebene hochgeleitet, auf der sich der gemeinsame Vorgesetzte der betroffenen Parteien befindet.“[64]

Gegen Einliniensysteme sprechen vor allem lange und umständliche Kommunikationswege. „Eindimensionale Organisationsformen stärken die Macht übergeordneter Instanzen, führen aber auch dazu, daß die Koordination und Konfliktlösung verlangsamt wird und möglicherweise durch Instanzen erfolgt, denen die sachliche Detailkenntnis fehlt.“[65] „Es entsteht das klassische Bild einer Pyramide mit einem kleinen Entscheidungszentrum an der Spitze und einer breiten Basis untergeordneter Mitarbeiter. [...] Die Koordination der einzelnen Arbeitsschritte erfolgt durch die Hierarchie, d.h. ein Heer von Vorgesetzten auf unterschiedlichen Rängen, denen gerade soviel Autorität delegiert wird, daß sie

[63] **[Picot1984]** S. 119
[64] **[Staehle1991]** S. 660
[65] **[Wild1973]** S. 47 Spalte 1

eine überschaubare Anzahl von Untergebenen kontrollieren können. Dem Ausmaß an delegierter Autorität, dessen Festlegung nur dem Top Management zusteht, entspricht die Bürde der Verantwortung für eine erfolgreiche Ausführung der Teilaufgabe.“[66]

Insgesamt erscheinen Einliniensysteme aufgrund starrer Hierarchie und aufwendiger Kommunikationswege für den Einsatz in dynamischen Umweltbedingungen nicht mehr zeitgemäß.

Es ist jedoch die Frage, ob Mehrliniensysteme in der Lage sind, eine ausreichende Flexibilität im Umgang mit der Umwelt zu schaffen. Befürworter von Mehrliniensystemen erwähnen vorrangig die Möglichkeit zur Vermeidung umständlicher, filternder und langsamer Hierarchie. Mehrliniensysteme sind allerdings nicht in der Lage, hierarchische Strukturen ganz zu umgehen. Eine Stelle bleibt formal immer noch auf die Kommunikation mit vorgesetzten Stellen beschränkt. Für dieses Problem wurden weitere Modelle entwickelt, wie z.B. die Projekt-Matrix Organisation von Mintzberg[67], in der sich Mehrlinienstrukturen nur für die Dauer von Projekten bilden. Dieses System ist aber nur bei Unternehmen anwendbar, die sich mit großen Projekten beschäftigen, wie z.B. im Schiffbau oder in der Raumfahrttechnik.

Außerdem verbessert eine funktionale Aufteilung die fachliche Kompetenz der Entscheidungsträger. Daraus resultiert eine **Kompetenzüberschneidung**, die durchaus beabsichtigt ist. Bei Konflikten muß eine argumentative Diskussion folgen, die von Stellen geführt wird, die dasselbe Problem in unterschiedlicher Sichtweise betrachten. Daraus resultiert angeblich eine höhere Qualität der Aufgabeninhalte[68]. Dieser Vorteil ist jedoch in der Praxis nicht in allen Situationen gegeben, da ein Mehrliniensystem auf hierarchischen Strukturen aufbaut. Es ist denkbar, daß die beteiligten Entscheidungsträger nicht nur im Sinne der Aufgabenerfüllung entscheiden, sondern auch eigene Interessen vertreten, die eine effektive Entscheidung behindern.

Eindeutiger Nachteil von Mehrliniensystemen ist, daß durch Aufgabenteilung und mehr Kommunikationswege schwerer ein Verantwortlicher für Fehlentwicklungen gefunden werden kann. Somit kann ein Entscheidungsträger u.U. darauf bauen, daß eine Entscheidung nicht ihm allein zur Last gelegt wird. Damit kann verantwortungsloses Verhalten begünstigt werden, sofern nicht eine gegenseitige Kontrolle der beteiligten Instanzen gewährleistet ist.

Matrix- und besonders Tensororganisationen sind wegen der Schwierigkeit graphischer Darstellung in der Praxis weniger geeignet. Eine konkrete Organisationsstruktur kann schwerlich das Werk eines einzelnen sein. Deshalb wird eine graphische Darstellung benötigt, die als Diskussionsgrundlage allen Beteiligten verständlich sein muß. Da viele Funktionsbeziehungen bestehen, wird eine Darstellung in einer einzigen Graphik unübersichtlich. Es müssen viele Teilsysteme geschaffen werden, deren Anzahl mit der Zahl der

---

[66] **[Staehle1991]** S. 660
[67] vgl. **[Mintzberg1979]** S. 172 f.
[68] vgl. **[Staehle1991]** S. 666

Stellen quadratisch (Matrixorganisation) oder kubisch (Tensororganisation) anwächst. Bei vielen Teilsystemen wird ein Verständnis der globalen Struktur erschwert.[69]

Zusammenfassend kann gesagt werden, daß Mehrliniensysteme Ansätze zu flexiblerer und in Teilaspekten kompetenterer Aufgabenerfüllung bieten. Die Vorteile sind aber nicht eindeutig, vor allem sind sie in der praktischen Umsetzung der Organisationsform nicht gewährleistet. Der entscheidende Mangel liegt in der Praxis am Festhalten an hierarchischen Strukturen und Denkmustern der beteiligten Personen.

Gegen hierarchische Systeme im allgemeinen sprechen folgende Aspekte:

Durch vorgeschriebene **Kanalisierung der Informationsströme** entsteht eine Beschneidung der Kommunikationsmöglichkeiten. „Soweit die Kommunikation an den Dienstweg gebunden ist, gibt die hierarchische Struktur die Kanäle und die Richtung der Informationsübertragung vor. Sie wirkt dabei stark restriktiv, schließt also andere Möglichkeiten der formalen Kommunikation aus und hat auf diese Weise erheblichen Einfluß auf das Informationspotential der einzelnen Stelleninhaber, auf die Informationsfilterung, -verstärkung u.a.. Dabei ist zu bedenken, daß dies [...] auch als Instrument der Machterhaltung und -ausübung benutzt wird."[70] Mit steigender Anzahl der Hierarchieebenen wird ein gefilterter Nachrichtenfluß eine schlechte Grundlage für Entscheidungen. Unvollkommene oder unzuverlässige Informationen hemmen die Kommunikation und somit die Möglichkeiten der Zusammenarbeit einzelner Stellen. Dann ist nicht auszuschließen, daß besonders im Forschungs- und Entwicklungsbereich Tätigkeiten doppelt durchgeführt werden.[71]

„Da Kommunikationswege hauptsächlich vertikal verlaufen, entsteht in den meisten Fällen eine ständige Überlastung der Instanzen mit Kommunikations- und Kontrollaufgaben. Von oben werden Ziele und Aufgaben vorgegeben, von unten wird Vollzug gemeldet."[72]

Hierarchie schafft organisationsinterne Konkurrenz und fördert damit strategisches Handeln der beteiligten Subjekte, das jedoch nicht zwangsläufig den Unternehmensinteressen dient.

Zusammenfassend läßt sich sagen, daß Liniensysteme aufgrund ihrer mehr oder weniger starren Hierarchien für dynamische Umweltsituationen nicht mehr zeitgemäß sind. Vorgeschriebene Kommunikationswege behindern den Informationsaustausch und damit eine umfassende Kooperation der Abteilungen. Dieser Nachteil ist auch bei der Mehrlini

[69] **[Blum1980]** S. 85 f.
[70] **[Wild1973]** S.47
[71] vgl. z.B. **[Lauterburg1980]**
[72] vgl. **[Staehle1991]** S. 660

enorganisation gegeben, die nicht auf feste Kommunikationsstrukturen verzichtet, sondern nur nach Funktionen aufgeteilte Alternativen bietet.
Im Rahmen dieser Arbeit soll ein Organisationsmodell entwickelt werden, das möglichst universell einsetzbar ist. Es muß sich durch Einfachheit und Transparenz der Darstellungsform auszeichnen. Deshalb ist die einfachere Struktur der Einlinienorganisation vorzuziehen. Sie kann in eingängiger graphischer Form, in einer Ebene dargestellt werden, und bietet dabei beliebige Aggregationsmöglichkeiten. Für das Organisationsmodell der Einliniensysteme bedeutet die Verwendung von Stäben keine große formale Erweiterung, kann aber eine gewisse Flexibilität in der Auslegung ansonsten starrer hierarchischer Beziehungen bringen.

3.2 Das Datenmodell

Kapitel 3.1.1 hat gezeigt, daß flüchtige Kooperationsstrukturen, wie z.B. eine Projektorganisation in Arbeitsgruppen, besonders effektiv angesichts dynamischer Umwelteinflüsse arbeiten. Andererseits können besonders größere Organisationsformen nicht ohne ordnende hierarchische Strukturen auskommen. Somit sollte ein Organisationsmodell diese zwei Organisationsstrukturen unterstützen.
Beide Aspekte können nicht gleichzeitig graphisch dargestellt werden, da die informellen Kommunikationsstrukturen der Projektorganisation meist zwischen vielen Abteilungen verlaufen. Wie in Kapitel 2.1.3 angedeutet, erlaubt moderne Bürokommunikationstechnik eine räumliche Trennung der Mitarbeiter eines Projektes. Doch dann ist eine hierarchische Struktur nicht in der Lage, alle an einem Projekt beteiligten Mitarbeiter gleichzeitig darzustellen.

Es wurde deshalb ein Organisationsmodell entwickelt, das eine Gestaltung von Projektstrukturen mit Hilfe von Arbeitsgruppen ermöglicht. Darüber hinaus gibt es eine konservativere Darstellung hierarchischer Teile einer Unternehmung durch das Stab-Linien-Modell.[73] Durch Verwendung beider Sichtweisen sind flexible, flüchtige Formen denkbar, deren Komponenten (bzw. Mitarbeiter) gleichzeitig durch eine fundierte hierarchische Struktur verwaltet werden können. Die Organisationsform kann von streng hierarchischer Ordnung mit gar keinen oder wenigen freien Arbeitsgruppen bis hin zu vollkommen offenen objektorientierten Strukturen variieren, in denen die hierarchische Ordnung nur noch zur verwaltungstechnischen Identifikation der Mitarbeiter dient. Das daraus resultierende Organisationsmodell erweitert das Organisationsmodell des GroupFlow - Systems um eine Möglichkeit zur Strukturierung von Arbeitsgruppen und eine Ausdehnung der Einlinienorganisation auf ein Stab-Linien-Modell.

Die einzelnen Elemente des Organisationsmodells sind **Personen**, **Abteilungen**, **Arbeitsgruppen** und **Rollen**. Auf diesen Grundelementen werden hierarchische Strukturen der Abteilungshierarchie oder Projektorganisation in Arbeitsgruppen aufgebaut. Sie stehen zueinander in folgenden Beziehungen:

[73] vgl. Kap. 3.1.3

Abbildung 3-7: Relationen der organisatorischen Grundelemente

Eine Erweiterung der Elemente ist möglich, muß aber im Kontext der GroupFlow Umgebung erfolgen. Das ist nötig, da die organisatorischen Einheiten im GroupFlow - Umfeld zum Erledigen bestimmter Teilaufgaben eines Geschäftsprozesses ausgewählt werden. Eine konzeptionelle Vorarbeit zur Erweiterung bietet z.B. **[Heinz1995]**. Der Autor beschreibt die Möglichkeit zum Einsatz von sog. Software-Agenten. Darunter versteht er Applikationen, die innerhalb von Geschäftsprozessen automatisch gestartet werden können („A task can not only be performed by a human actor or an organizational unit but also automatically by a software agent.[74]“
Nach Absprache mit der Firma Pavone stellen Software-Agenten (bisher) keine eindeutig aufbauorganisatorischen Elemente dar und werden bis auf weiteres aus der Aufbauorganisationsdatenbank herausgelassen.

In der Abbildung 3-7 ist dargestellt, daß alle weiteren Grundelemente von Personen abhängen. Arbeitsgruppen, Abteilungen und Rollen werden ihnen zugeordnet und bekommen damit eine organisatorische Funktion. Die Zuordnungen beschreiben Fähigkeiten, Stellung und Zugehörigkeit der Personen zu anderen Einheiten. Im Datenmodell wird die Zugehörigkeit einer Person zu anderen Einheiten nur im betreffenden Personenobjekt gespeichert, nicht jedoch in den anderen Objekten. Die Mitglieder einer Abteilung werden also nicht in der Abteilung explizit aufgeführt, sondern können nur durch eine Abfrage aller Personen bestimmt werden. Diese Redundanz in der Datenhaltung ermöglicht eine einfachere Verwaltung durch kleinere Datenbestände, verursacht aber höheren Aufwand bei der Zuordnung.

In den folgenden Unterkapiteln werden die einzelnen Grundelemente näher erläutert und deren Relationen in einer modifizierten Form von ‘Entity-Relationship-Modellen’

[74] **[Heinz1995]** S. 23

beschrieben. Kapitel 3.2.2 stellt das Konzept der Rolle vor, mit dem konkrete Stellen variabel identifiziert werden können.

3.2.1 Stellen

Eine Stelle wird durch ihren Namen und ihre Ausprägungsform identifiziert. Außer für Abteilungen ist keine eindeutige Hierarchie von Stellen definiert. Deshalb kann der Name von Personen und Arbeitsgruppen nur einmal vergeben werden. Abteilungen unterschei-
den sich hingegen auch durch ihre hierarchische Zuordnung. Es kann also Abteilungen mit gleichem Namen geben; jedoch können zwei Abteilungen mit demselben Namen nicht derselben Abteilung angegliedert werden.

Die Abteilungshierarchie gilt als formale Abbildung des Unternehmens. Zur einfachen Identifikation der Personen im Modell müssen sie immer einer Abteilung formal zuge-
ordnet sein. Auf eine Zuordnung von Arbeitsgruppen auf Abteilungen wurde verzichtet, da dies eine informelle Einschränkung der Zielsetzung der Arbeitsgruppen bedeuten könnte.

Die Entscheidung, Arbeitsgruppen **keiner** Abteilung formal zuzuordnen oder anzuglie-
dern ist jedoch nicht eindeutig vorteilhaft. So kann es Probleme mit der Ressourcenaus-
stattung der Arbeitsgruppen geben, wenn andere Einheiten Kapazitäten abgeben sollen. Andererseits kann es **bei** einer Angliederung einer Arbeitsgruppe an eine spezielle Abteilung vorkommen, daß sie nur über Ressourcen der Stammabteilung verfügen darf und sich andere Abteilungen gegen eine Benutzung ihrer Ressourcen sperren.
Daraus entsteht ein Problem in der praktischen Umsetzung des Organisationsmodells: Während einerseits eine ausreichende Ressourcenausstattung der relativ unabhängigen Arbeitsgruppen gewährleistet sein muß, darf andererseits für hierarchische Abteilungen nicht der Zwang entstehen, zu viele Ressourcen freigeben zu müssen. Das organisatori-
sche Modell sieht keine ordnenden Instanzen für diese Kapazitätenverteilung vor. Wenn also ein Mischmodell zwischen Projektorganisation und Abteilungshierarchie realisiert werden soll, so ist unternehmensinterne Überzeugungsarbeit zu leisten, damit einzelne Stellen das Unternehmensziel über ihre lokale Zielerfüllung stellen.

3.2.1.1 Personen

Die einfachste und konkreteste organisatorische Einheit ist eine Person. Sie besitzt ein Bündel von Fähigkeiten und Zugehörigkeiten zu anderen Einheiten. Wie in Abbildung 3-7[75] dargestellt, hängen alle anderen organisatorischen Einheiten von Personen ab, d.h. eine Person kennt ihre Zugehörigkeit zu einer Abteilung, Arbeitsgruppe oder Rolle, diese aber nicht ihre Mitglieder oder Inhaber.

---

[75] Kap. 3.2

Die folgende Tabelle stellt eine etwas abgewandelte Form eines **Entity-Relationship-Modells** dar. Ein Entity-Relationship-Modell ist eine tabellarische Darstellungsweise der Relationen eines Grundelementes mit anderen Elementen sowie seiner Attribute. In der linken Spalte steht der Name der organisatorischen Einheit (engl.: Entity). Die mittlere Spalte kennzeichnet die Relationsmöglichkeiten zu anderen organisatorischen Einheiten und - in Klammern - die Anzahl der Verbindungsbeziehungen. Die dritte Spalte führt alle Attribute der Einheit auf.

Einheit	**Relationen** (engl.: relations)	**Attribute** (engl.: Attributes)
Person	Abteilung (n:1) Abteilungsleiter (1:1) Mitglied in Arbeitsgruppen (n:m) Manager in Arbeitsgruppen (n:m) Rollen (n:m)	**Vorname** **Mittel-Initial** (1 Buchstabe) **Nachname** **Certifier**[76] **Abteilung** **Ist Abteilungsleiter** (ja oder nein) **Gehört zu Arbeitsgruppen** **Manager in Arbeitsgruppen** **Rollen**

Tabelle 1: Relationen und Attribute einer Person

Eine verbale Übersetzung der Relationen könnte lauten:
Eine Person
- gehört zu genau einer Abteilung,
- kann in mehreren Arbeitsgruppen sein,
- kann mehrere Rollen besitzen bzw. die in der Rolle beschriebene Befähigung haben,
- kann Manager höchstens einer Abteilung sein,
- kann Manager in mehreren Arbeitsgruppen sein,
- kann mehrere Rollen innehaben.

3.2.1.2 Abteilungen und Stabstellen

Eine Abteilung kann aus Personen und Unterabteilungen bestehen. In der Einlinienhierarchie hat eine Abteilung höchstens eine vorgesetzte Abteilung.

In folgender Tabelle wurden diejenigen Attribute weggelassen, die nicht explizit gespeichert werden. Mitglieder, der Manager und untergeordnete Abteilungen werden nicht formal als Attribute einer Abteilung abgelegt, sondern müssen durch entsprechende Attribute der betreffenden Einheiten bestimmt werden. Weitere berechenbare Attribute

[76] Das Attribut ‘Certifier’ ist ein Anhängsel für den Namen einer Person und ist als Kurzform für eine organisatorische Zuordnungen gedacht. Beispiel: Eine Person gehört zur deutschen Tochterfirma ‘Rita Pavone’ eines internationalen Konzerns und erhält den Certifier ‘PAVONE/DE’.

sind direkte und indirekte Kapazität, also die Anzahl der Mitarbeiter nur der Abteilung oder zusätzlich der Unterabteilungen.

Einheit	Relationen	Attribute
Abteilung	Personen als Mitglieder (1:m) Personen als Manager (1:1) Ober-Abteilung (m:1) Unter-Abteilung (1:m)	**Beschreibung** **Name** **Hierarchische Ebene** **Übergeordnete Abteilung** **Ist Stabstelle** (ja oder nein)

Tabelle 2 Relationen und Attribute einer Abteilung

Dabei gelten folgende verbale Regeln:
- Eine Abteilung muß einen Manager haben.
- Eine Person kann nur zu einer Abteilung gehören.
- Deshalb kann eine Person nur Manager einer Abteilung sein.
- Eine Abteilung hat nur eine direkte Überabteilung.
- Sie kann aber mehrere Unterabteilungen haben.

Auf **Stellenschlüssel** wurde verzichtet, da es in der Organisationstheorie keine normierte Darstellung dafür gibt. Stellenschlüssel sind eine Kurzform, aus der die Rangstufe, das Aufgabengebiet und der Typ einer Stelle ersichtlich sein können. Eine Ausprägung von Stellenkennziffern kann z.B. eine ebenenweise Numerierung sein. Dabei bekommen die Unterabteilungen einer Abteilung deren Stellenschlüssel und eine angehängte fortlaufende Numerierung.[77]

Abbildung 3-8 Stellenkennziffern

[77] Vgl. Abbildung 3-8

Da das Organisationsmodell dieser Arbeit auf eine praktische Verwendung im Rahmen einer Workflow - Umgebung hinarbeitet, erscheinen Stellenschlüssel auch aus zwei anderen Gründen nicht zweckmäßig:
Ein Modell kann wie in Abbildung 3-8 Stellenschlüssel einer bestimmten Position automatisch zuordnen. Wird jedoch eine Stelle gelöscht, oder umgesetzt, so ändert sich u.U. eine ganze Reihe von Schlüsseln. So hat beispielsweise in Abbildung 3-8 ein Löschen der Abteilung 1.1.2 Folgen für die Abteilungen rechts von ihr. Somit erscheint eine automatische Zuordnung nur als Orientierungshilfe innerhalb graphischer Präsentationen zweckmäßig. Zur Navigation in organisatorischen Strukturen wurden im Rahmen der prototypischen Umsetzung jedoch mächtigere Werkzeuge geschaffen. Das ist die in Kap. 4.4.2.1.3[78] beschriebene Funktion zum Verkleinern und Vergrößern der Darstellung und eine Möglichkeit zum Kompaktieren[79].

Eine Verwendung von festen (nicht automatisch zugeordneten) Stellenschlüsseln ist innerhalb dieser Arbeit ebenfalls nicht vorgesehen, da sie keine verläßliche Orientierungshilfe in der Abteilungshierarchie bieten. Es besteht allerdings die Möglichkeit zur Vergabe von Stellenschlüsseln in der textuellen Beschreibung einer Abteilung.

3.2.1.3 Arbeitsgruppen

Arbeitsgruppen und deren Attribute wurden direkt aus dem *Lotus Notes* - Adreßbuch (Name & Adress Book) übernommen. Diese Beziehung ist bis in die Benennung der Feldvariablen der Organisationsdatenbank einheitlich und soll eine spätere Kompatibilität der Datenbanken gewährleisten[80].

Arbeitsgruppen werden aus Personen und anderen Arbeitsgruppen zusammengestellt. Dabei sollte es, außer in Definitionsphasen, keine leeren Arbeitsgruppen geben. Außerdem ergibt eine Arbeitsgruppe mit nur einer Person keinen Sinn, da ansonsten kein Unterschied zu einer Personenstelle besteht.

Personen werden meist aus verschiedenen Abteilungen für einen bestimmten Zeitraum in einer Arbeitsgruppe zusammengefaßt. Eine Zuordnung von Arbeitsgruppen auf spezielle Abteilungen wird nicht vorgenommen, da deren Zielsetzungen durch Wahl einer vermeintlich dominierenden Abteilung beeinflußt werden könnten. Darüber hinaus wäre eine Arbeitsgruppe evtl. nur durch eine Person in der betreffenden Abteilung vertreten, und ein Wechsel dieser einzelnen Person würde eine aufwendige formale Umorientierung der Arbeitsgruppe erforderlich machen.

Anders als bei der formalen Hierarchie der Abteilungen besteht keine Hierarchie bei der Einordnung von Arbeitsgruppen. Sie stehen in einem mehr informellen Verhältnis, da eine Arbeitsgruppe vielen anderen Arbeitsgruppen zuarbeiten kann. Somit entsteht aus

---

[78] ‘Formatierfunktionen und deren Verwendungsmöglichkeiten’
[79] vgl. Kap. 4.4.2.1.1
[80] vgl. Kap. 4.3.2 ‘Konvertierungen beim Laden und Speichern’

der Summe der Relationen eine Art Netzwerk von Arbeitsbeziehungen, die aber keine Rangreihenfolge ausdrücken sollen.

In der nachfolgenden Tabelle fehlen die Attribute 'zugeordnete Abteilungen', 'Mitglieder', 'Manager' und 'Kapazität' (Anzahl der Mitarbeiter), die aus entsprechenden Attributen der anderen Einheiten ermittelt werden.

Einheit	Relationen	Attribute
Arbeitsgruppe	Person als Mitglied (n:m) Person als Manager (1:m) Arbeitsgruppe (n:m)	**Beschreibung** **Name** **Ist Arbeitsgruppen zugeordnet**(Liste)

Tabelle 3: Relationen und Attribute einer Arbeitsgruppe

Dabei gelten folgende verbale Regeln:
- Eine Arbeitsgruppe kann, muß aber keinen Manager haben.
- Ein Manager kann mehrere Arbeitsgruppen leiten.
- Personen können zu mehreren Arbeitsgruppen gehören.
- Eine Arbeitsgruppe kann Unterarbeitsgruppen beinhalten.
- Eine Arbeitsgruppe kann Bestandteil von mehreren anderen Arbeitsgruppen sein.

3.2.2 Das Rollenkonzept

Das Rollenkonzept ist in **[Heinz1995]** eingehend beschrieben. An dieser Stelle wird die Funktion von Rollen aus organisatorischer Sicht erarbeitet, um zu zeigen, daß eine besondere Ausprägung von Rollen, die *relationalen Rollen,* wahrscheinlich einfacher durch andere Ausprägungen von Rollen substituiert werden können.

Das Rollenkonzept geht auf eine Vorüberlegung aus dem Bereich der Ablauforganisation zurück:

Die Ablauforganisation ist eine *formale Struktur,* die Regeln für *real ablaufende* Prozesse festlegt. Das ist z.B. in der Abbildung 3-9 der Fall. Dort sind in der Ablauforganisation formale Strukturen[81] für Geschäftsprozesse vom Typ **D** und **E** definiert. Die formale Ablaufstruktur **E** wird von den realen Prozessen **E1** und **E2** durchlaufen, d.h. die realen Geschäftsprozesse **E1** und **E2** befinden sich zu einem *Zeitpunkt* an einer Stelle eines Gesamtprozesses vom Typ **E**.[82]

---

[81] hier Schablonen genannt
[82] zur näheren Erläuterung der beteiligten Datenbanken sei auf **[NastHilp1994a] [NastHilp1994b]** verwiesen

Abbildung 3-9 Geschäftsprozesse in Abhängigkeit zur Ablauforganisation

Eine Ablaufstruktur muß flexibel konzipiert sein, da sie den Anforderungen möglicherweise. sehr verschiedenartiger Umweltsituationen, also den unterschiedlichsten Ausprägungen eines Geschäftsprozesses genügen soll.

Bei der Entwicklung einer Ablaufstruktur, also dem formalen Ablauf eines Geschäftsprozesses, ist es oftmals zu restriktiv, eine spezielle Aufgabe einer bestimmten Stelle zuzuordnen. Mit wachsender Komplexität der Infrastruktur und der Prozesse wird ein von konkreten Personen abstrahierendes, flexibles Rollenkonzept benötigt, das eine relative Zuordnung zu Personen, Abteilungen und Arbeitsgruppen erlaubt. Hierfür sprechen hauptsächlich zwei Aspekte: Zum einen kann bei dynamischer Aufbaustruktur die Zusammensetzung und der Name einer Stelle schnell geändert werden - evtl. wird eine Stelle sogar aufgelöst. Zum anderen ist es hilfreich, eine abstrakte Stelle zu definieren, die dann unter den realen Umständen eines Geschäftsprozesses einer realen Stelle zugewiesen werden kann.[83]

Beispiel:

Gegeben sei ein Geschäftsprozeß ‘Reisekostenabrechnung’. Dieser wird von einer Person initiiert, die eine Geschäftsreise unternommen hat. Im Verlauf des Geschäftsprozesses muß dann der unmittelbare Vorgesetzte die Abrechnung genehmigen. Eine Abteilungssekretärin archiviert die Abrechnung, usw..

Solch ein Geschäftsprozeß muß möglichst universell gehalten werden, damit er von jedem Mitarbeiter des Unternehmens benutzt werden kann. Daher sind in der formalen Ablaufstruktur keine bestimmte Sekretärin oder ein expliziter Vorgesetzter definiert,

---

[83] vgl. **[Heinz1995]**, S. 22 f.

sondern sie werden abstrakt durch Rollen angegeben. Ruft dann ein Mitarbeiter diesen Prozeß auf, wird aus dem jeweiligen realen Kontext die zuständige Sekretärin und der betreffende Vorgesetzte eingesetzt.

Im Umfeld der *GroupFlow* Produkte gibt es drei verschiedene Ausprägungen von Rollen:
- die einfache Rolle,
- die Rolle mit Parameter,
- die relationale Rolle.

Eine einfache Rolle ordnet einer Person Fähigkeiten oder Befugnisse zu, die es ihr erlauben, bestimmte Arbeiten durchzuführen. Eine parametrisierte Rolle beschreibt verschiedene Ausprägungen einer Klasse von Rollen, ist also eine Spezialisierung einer einfachen Rolle. Desweiteren kann eine Rolle eine Beziehung zwischen Stellen beschreiben, z.B. ‘Sekretärin einer Abteilung’.

Beispiel:
Eine einfache Rolle kann die Eigenschaft ‘Programmierer’ beschreiben. Zu diesem globalen Typ von Rolle kann eine Differenzierung in verschiedene Programmiersprachen sinnvoll sein. So kann die allgemeine Rolle ‘Programmierer’ für Programmierer benutzt werden, die irgendeine Programmiersprache beherrschen. Die Rolle ‘Programmierer Ada’ beschreibt hingegen Programmierer, die eine Applikation in Ada erstellen können. Somit ergeben sich verschiedene Spezialierungsebenen der Rolle.

Abbildung 3-10 Mögliche Ausprägungen eines Rollentyps

In der GroupFlow Umgebung sind **relationale Rollen** bisher nicht realisiert worden, es gibt nur ein Konzept für ihren Einsatz.[84] Sie sind eine technische Zuordnungsvorschrift, die ohne menschliches Zutun eine automatische Zuordnung einer abstrakten Rolle zu einer realen Stelle vornehmen sollen. Eine relationale Rolle schafft somit z.B. die Möglichkeit, eine zuständige Sekretärin maschinell zu bestimmen. Dabei ist in den meisten Fällen Kenntnis über interne Struktur anderer Datenbanken nötig.[85] Deshalb wird im Datenmodell auf relationale Rollen verzichtet, da es in den meisten Fällen einfacher und sicherer erscheint, eine situationsbezogene Zuordnung von menschlichen Bearbeitern vornehmen zu lassen. Deshalb ergibt sich keine große Einschränkung, weil automatische Berechnungen auf relationalen Daten nicht immer ein Ergebnis liefern, muß das Konzept relationaler Rollen ohnehin eine Ausnahmebehandlung durch menschliche Bearbeiter vorsehen.

3.2.3 Das Datenmodell der Rollen

Das PAVONE GroupFlow - Konzept ist dokumentorientiert, d.h. es werden hauptsächlich ‘Dokumente’ von Stelle zu Stelle weitergeleitet (engl.: Routing). Jedes Dokument erscheint meist am PC einer Person in möglicherweise recht verschiedenartigen Ausprägungen (z.B. in akustischer, graphischer, textueller Form oder Mischungen daraus). Damit ist im Endeffekt immer eine mehr oder weniger spezifizierte Person die ausführende Stelle. Aus diesem Grund werden Rollen bisher nur Personen zugewiesen.

Eine Rolle ist eine Abstrakion einer Stelle, die zur Laufzeit eines Geschäftsprozesses zur Bestimmung einer konkreten Stelle herangezogen wird. Das geschieht folgendermaßen: Die Bearbeitung einer Aufgabe erfordert eine bestimmte Eigenschaft oder Befähigung (also eine Rolle). Zur Laufzeit eines Geschäftsprozesses werden dann aus der potentiellen Menge der Bearbeiter Personen ausgewählt, die für diese Rolle zugelassen sind.
Für diese Funktionalität muß die Aufbauorganisationsdatenbank Mittel zur Verwaltung von Rollen bereitstellen und die Möglichkeit der Zuweisung auf bestimmte Stellen bieten.

Rollen verschiedener Typen müssen sich im Namen unterscheiden. Sind sie vom selben Typ, so dürfen sie keine gleichen Parametern besitzen.

Rollen haben folgendes Entity-Relationship-Modell. Die Rolleninhaber werden aus Personenobjekten ermittelt.

---

[84] vgl. **[Heinz1995]**, S. 27 ff.
[85] Ebenda, S. 27

Einheit	Relationen	Attribute
Normale / parametrisierte **Rolle**	Person (n:m)	Beschreibung Name Parameter (evtl. nicht vorhanden)

Tabelle 4 Relationen und Attribute einer Rolle

Dabei gelten folgende verbale Regeln:
- Eine Rolle kann mehreren Personen zugeordnet werden.
- Eine Person kann mehrere Rollen innehaben.

3.3 Graphische Darstellung des Modells

Die graphische Präsentation von Organisationsstrukturen ist zumindest so alt wie die Organisationstheorie selber, da der Mensch eine Theorie am einfachsten anhand bildlicher oder plastischer Artefakte mitteilen kann. Das gegenseitige Verstehen ist nötig, um eine gemeinsame Diskussionsgrundlage zu bekommen. Ohne Diskussion, Argumentation und vergleichende Bewertungen hätten nicht solch leistungsfähige Unternehmensformen entwickelt werden können, wie sie heute existieren.[86] Wenn von Organisationsformen die Rede ist, denken die meisten wirtschaftlich gebildeten Menschen an eine graphische Darstellung, ein **Organigramm**.

Es wurden vielfältige Formen entwickelt, mit der Organigramme zunächst auf Papier präsentiert werden konnten. Mit gestiegenen Fähigkeiten bildete dann der Computer ein effizientes ‘Werkzeug’ zur Vorbereitung von Präsentationen. Besonders im Bereich der Endbenutzerwerkzeuge sind unter dem Stichwort **Multimedia** vielfältige Applikationen verfügbar, mit der Präsentationen auch direkt, z.B. über Projektoren, mit Lautsprechern u.ä., gezeigt werden können.

Es steht der nächste Schritt an, der im interaktiven Umgang mit den graphischen Repräsentationen besteht:
Zunächst bedeutet graphische Abbildung schlicht eine Erleichterung im Umgang mit dem Computer. Darüber hinaus wird über das Bild das Auge, das am meisten beanspruchte und somit trainierteste Sinnesorgan, voll in den Dialog mit dem Computer einbezogen. Es entsteht eine neue Rollenverteilung zwischen Mensch und Computer, die Veränderungen auf beiden Seiten mit sich bringt: - für den Benutzer eine neue Vorstellungswelt mit heutzutage bei weitem noch nicht ausgereizten Möglichkeiten, - für den Computer ein neuer Aufgabenbereich, in dem sich seine Leistungsfähigkeit fortlaufend und immens steigern muß, um mit den Ansprüchen schritthalten zu können. So kann z.B. die

[86] vgl. **[Raeithel1991]** S. 411 ff. Kap. ‘Symbolic models for theoretical work’

Darstellung einer künstlichen Welt[87] mit beliebiger Detailtreue ausgestattet werden, damit aber auch den leistungsfähigsten Rechner leicht an seine Leistungsgrenzen bringen.

Der erste Abschnitt dieses Kapitels[88] befaßt sich in abstrakter Form mit den Grenzen und Gefahren einer vom Computer erzeugten Vorstellungswelt.

Da für die Darstellung von organisatorischen Strukturen multimediale Techniken wenig Sinn machen oder in diesem Rahmen zu aufwendig sind, wird im zweiten Abschnitt[89] untersucht, welche graphischen Möglichkeiten bestehen.

3.3.1 Möglichkeiten und Grenzen von 'Modellwelten'

Die Bildliche Darstellung eines Modells ist immer an räumliche Strukturen gebunden, die im Falle der Bildschirmdarstellung zwei- oder angedeutet dreidimensional sein können. Diese Eigenschaft kann man sich vor Augen führen, wenn man bedenkt, daß sowohl die Objektorientierung, die Schreibtischmetapher der heutigen graphischen Betriebssysteme, als auch die Hypertextidee eines gemeinsam haben: sie bevorzugen räumlich sichtbare Strukturen und vernachlässigen zeitliche, interaktive Abläufe.

Die Aufgabe der visuellen Wahrnehmung durch den Benutzer besteht darin, aus Bildern zuverlässig Eigenschaften der 'Welt' abzuleiten, die durch diese Bilder abstrahiert wird. Graphische Modelle sind Repräsentationen, mit deren Hilfe man eine oder auch möglichst viele Eigenschaften maschinell simulieren will. Problempunkte, die es bei der Gestaltung eines graphischen Interaktionssystems zu berücksichtigen gilt, sind in etwa folgende:

- Werden ausreichend viele Verhaltensweisen der externen Welt dargestellt?
- Werden sie gut genug simuliert?
- Wird zu sehr abstrahiert, daß die ursprüngliche, reale Situation nicht mehr erkennbar ist?
- Werden zu viele Verhaltensweisen simuliert, die von den wesentlichen Eigenschaften ablenken und den Prozeß des Verstehens beeinflussen?
- Werden zu viele Verhaltensweisen simuliert und damit die Antwortzeiten des Systems zu groß?
- Wird mit einer Auswahl der dargestellten Informationen und Reaktionen eine subjektive Abstraktion gemacht, die einen objektiven Umgang behindert?

Zu dieser Aufzählung können noch viele Aspekte hinzugefügt werden, jedoch zeigt bereits dieser Ausschnitt, daß ein optimales graphisches Interaktionsmodell nicht realisierbar ist. Selbst wenn es zu einem Zeitpunkt nahezu optimal ist, kann es zu einem späteren Zeitpunkt nur noch ein Zerrbild der Realität sein. Um hundertprozentig optimal arbeiten zu können, muß in erster Linie die externe Repräsentation, also die reale

---

[87] engl.: Virtual Reality
[88] Kap. 3.3.1
[89] Kap. 3.3.2

Umwelt, betrachtet werden, um festzustellen, aufgrund welcher Eigenschaften welche Handlungsmöglichkeiten und Handlungseinschränkungen gegeben sind. Das beste Modell ist also die Realität selber, und es muß fortwährend beachtet werden, daß ein graphisches Modell keinen vollständigen Ersatz dafür leisten kann.

Darüber hinaus erfordert eine mit Interaktionsmöglichkeiten ausgestattete graphische Darstellung nicht nur strukturbezogene, sondern auch handlungsbezogene Konsistenz. Die realen Situationen und Situationsübergänge, für die das Modell gelten soll, sind in den meisten Fällen von unendlich vielen Umwelteinflüssen abhängig. Deshalb kann ein Modell als Abbild der Umweltsituationen nicht in allen möglichen erreichbaren Ausprägungen vorab getestet werden. Desweiteren kann nicht mit Ingenieurmethoden eine mathematische Zuverlässigkeit abgeleitet werden, da mathematische Formeln ein gewisses Maß an Unsicherheit der Umwelteinflüsse berücksichtigen müßten.
Wenn sich ein Modell dem Test aller möglichen Handlungskonstellationen verschließt, aber auch keine berechenbar große Zuverlässigkeit bietet, muß es als potentiell unzuverlässig gelten.
Ein graphisches Modell kann demnach nur bis zu einem gewissen Grad zuverlässig sein. Dessen Korrektheit und Zielerfüllungsgrad läßt sich nur über Erfahrungswerte mit dem Umgang des Systems erschließen, und diese Methode ist alles andere als verläßlich.[90]

Die großen Vorteile graphischer Modelle bestehen in der Möglichkeit, viele Informationen und Zusammenhänge mit relativ wenigen Objekten darzustellen. So steht in jedem Interaktionsschritt ein **Bildschirmobjekt** und seine Manipulation im Vordergrund der Betrachtung und nicht **Funktionsbereiche** oder **Editiermodi** in einer textorientierten Dateneingabemaske, die Zusammenhänge nur durch den Akt der Interaktion des Benutzers erkennen lassen. „Erst mit sichtbaren Strukturen ist es einem Benutzer möglich, sich die Interaktionsmöglichkeiten wie auch die mit einem Interaktionsschritt verbundenen Effekte, die sich ja als Strukturveränderungen wiederum auf das weitere Interaktionsverhalten auswirken (können), in einem größeren Umfang zu erschließen und damit auch komplexere Strukturen intellektuell beherrschbar zu machen. Die technischen Mittel zur Realisierung räumlicher Repräsentationen sind Bild- und Graphik-Repräsentationen am Bildschirm, die es ermöglichen, graphische Objekte als Datenstrukturen zu behandeln.“[91]

Graphische Interaktion bringt bereits in kleinen Details Vorteile. Es sind bereits wesentliche Grundlagen eines interaktiven Systems gelegt, wenn graphische Elemente nicht nur gezeichnet, sondern auch manipuliert werden können. Die visuelle Struktur aus Bildpunkten wird zu einem veränderbaren Objekt, in der die Interaktion mit einem Objekt jeweils eine Interaktion mit der Struktur als ganzem bewirkt. So ist es z.B. möglich, im Gegensatz zu einer lediglich gezeichneten (gemalten) Struktur ein Objekt räumlich zu verschieben, ohne daß die Graphik vom Benutzer formal ausradiert und an anderer Stelle neu gezeichnet werden muß.

---

[90] „Bisher hat alles anscheinend gut funktioniert, wir hinterfragen also nicht weiter und verlassen uns ab jetzt auf unser Modell“
[91] **[KeiSl1990]** S. 54

Desweiteren bieten bildhafte Darstellungen die Möglichkeit einer räumlichen Repräsentation, in der Objekte einer Struktur gleichzeitig dargestellt werden. Funktionelle Eigenschaften sind nicht sichtbar, sondern werden als Konsequenz der Interaktionen des Benutzers ausgeführt. Dadurch ist die Darstellung übersichtlich und überläßt dem Benutzer die Entscheidung, welche Aktionen er in welcher Reihenfolge ausführen möchte. Sie befreit den Benutzer von erzwungen sequentiellen Vorgangsschritten, die z.B. beim Bearbeiten von Eingabemasken eines Datenbanksystems negativ auffallen. „Das Phänomen der Interaktivität [ist] grundsätzlich mit der Orientierung auf die *individuelle Unterstützung* geistiger Arbeit verknüpft [...] - ausdrücklich nicht mit ihrer Ersetzung.“[92]

Daher reicht es nicht aus, eine detailgetreue Abbildung der Realität zu schaffen. Ein Modell muß sich daran orientieren, mit welchen Lernprozessen der Mensch als Benutzer sich den Sinn und die Abhängigkeiten der einzelnen Teilstrukturen erschließt. Nur das eigene in-Beziehung-setzen, also die Interaktion mit dem Modell erschließt ihm Zusammenhänge zwischen einzelnen Zuständen und damit die zeitlich ablaufenden Strukturen. Dieses Verständnis muß erweckt werden, um eine sinnvolle Interaktion mit dem Modell und eine ausreichende Akzeptanz zu gewährleisten.

3.3.2 Darstellungsformen des Organisationsmodells

Bei der bildlichen Darstellung sind einige Vorgaben zu beachten:
Zum einen läßt die rechteckige, relativ kleine Bildschirmfläche nur eine begrenzte Auswahl von Darstellungsformen zu. Desweiteren gibt es graphische Modelle, die nur eine begrenzte Leistungsfähigkeit besitzen, also nicht alle Konstellationen oder Größenordnungen zulassen. Da Repräsentationen mit Funktionalitäten zur Interaktion ausgestattet werden sollen, steht insbesondere die Berechenbarkeit bzw. schnelle Berechenbarkeit der Darstellung im Vordergrund.

3.3.2.1 Darstellung einzelner Elemente

Zur Darstellung von organisatorischen Elementen gibt es zwei Gestaltungsmerkmale. Zum einen ist das die **äußere Form**, welche entscheidend dafür ist, wieviele der Elemente gleichzeitig dargestellt werden können. Zum anderen gibt es eine **innere Aufteilung** d.h. eine Anordnung der Symbole oder Informationen im Element. Beide Merkmale sind voneinander abhängig, da die innere Aufteilung die Größe beeinflußt und die äußere Form und Größe die Möglichkeiten der inneren Gestaltung vorgeben.

---

[92] **[KeiSl1991]** S. 1

3.3.2.1.1 Abteilungen

Zunächst sollte eine äußere Form gewählt werden, die eine möglichst große Innenfläche hat. Gleichzeitig darf sie selbst aber nicht zu groß sein, um die Darstellung möglichst vieler Elemente zu ermöglichen. Die Form der meisten Bildschirme ist ein flaches Rechteck. Außerdem sind Textzeilen und Sinnbilder i.d.R. auch rechteckig.

Beispiel:

Abbildung 3-11: Verschiedene Abteilungsformen

Beide Formen sind gleich breit. Da aber das Rechteck flacher ist, können mehr Rechtecke auf einem Bildschirm angeordnet werden als Dreiecke.
Zudem bietet die rechteckige Form Platz für drei Informationen, das Dreieck bei gleicher Schriftgröße aber nur für zwei, zudem kürzere.

Als Grundform bietet sich somit ein Rechteck an. Nur wenn es zwingend nötig ist, einen offensichtlichen Unterschied zu verdeutlichen, sollten andere Formen hinzugezogen werden.
Im bisherigen Organisationsmodell gibt es nur zwei Typen, nämlich normale Stellen und Stabstellen. Dafür wird in der Organisationstheorie meist eine unterschiedliche Form verwendet.[93] In der Literatur wird eine weitere Unterscheidung in leitende und ausführende Stellen vorgeschlagen.[94] Dieses Merkmal ist aber nicht eindeutig, da auch übergeordnete Stellen ausführende Tätigkeiten übernehmen können. Letztlich sind nur die Stellen reine ausführende Stellen, die keine untergeordnete Abteilungen besitzen. Die sind im Organigramm ohnehin leicht erkennbar, da sie nur übergeordnete Linien (im graphischen Sinne) haben, deshalb erscheint eine Unterscheidung von Abteilungen und Stabstellen ausreichend.

Egon Blum gibt weitere Gesichtspunkte an, die theoretisch auch für die Wahl anderer Grundformen sprechen können:
„- Zweck der Darstellung
 - Anzahl der Stellen

---

[93] Stellen als Rechtecke, Stabstellen als abgerundete Rechtecke
[94] vgl. **[Kosiol1976]** S. 173

- Differenzierungsbedürfnis nach Stellenarten
[...]
- Übersichtlichkeit und Ästhetik des Schaubildes
- Erstellungs- und Reproduktionsaufwand."[95]

Farbe:
Blum schließt farbliche Differenzierungen aus, da sie einen zu hohen Reproduktionsaufwand hätten.[96] Das ist heutzutage kein Problem mehr. Farben könnten dazu verwendet werden, eine Teilmenge der Stellen einheitlich zu markieren. Die einzig nötige Unterteilung zwischen normalen Abteilungen und Stabstellen wird schon an der Form ersichtlich. Somit ist nur eine Farbe nötig, die rein ästhetischen Zwecken dienen kann.

Aus den Überlegungen sind zwei Formen für Abteilungen entstanden, die in Abbildung 3-12 zu sehen sind.[97]

Innere Aufteilung:

Eine Abteilung hat folgende direkte oder ermittelbare Attribute[98]:
- Beschreibung,
- Name,
- Hierarchische Ebene,
- Übergeordnete Abteilung,
- Mitglieder,
- Anzahl der Mitglieder,
- Abteilungsleiter,
- Ist Stabstelle.

Eine Beschreibung würde zuviel Platz beanspruchen. Hierarchische Ebene und übergeordnete Abteilung ergeben sich aus der graphischen Darstellung. Stabstellen haben eine andere Form. Deshalb sind nur folgende Merkmale relevant: Name, Mitglieder, Anzahl der Mitglieder, Abteilungsleiter. Für eine Auflistung aller Mitglieder ist nicht genug Platz vorhanden.
Desweiteren wurde im Rahmen dieser Arbeit mit Stellenschlüsseln experimentiert.[99] Durch den weiteren Platzbedarf der Stellenschlüssel wurden Abteilungsobjekte jedoch sehr groß. Aus den in Kap. 3.2.1.2 genannten Abwägungen zwischen Universalität des Modells und Berechenbarkeit von Stellenschlüsseln ist auf Stellenschlüssel später verzichtet worden.

Es ergibt sich folgende interne Aufteilung mit entsprechenden Attributen:

---

[95] **[Blum1980]** S. 42
[96] vgl. ebenda
[97] links normale Stelle, rechts Stabstelle
[98] vgl. Entity-Relationship Modell in Kap. 3.2.1.2
[99] vgl. 3.2.1.2

Name der Stelle
Anzahl d. Mitglieder
Poststelle 2
Charly Brown
Statistik 7
Joe Sloppy
Abteilungsleiter

Abbildung 3-12 Abteilungen in graphischer Form

Bemerkung:
In der prototypischen Umsetzung des Organisationsmodells haben Abteilungen abhängig von der Größe der Darstellung verschiedene innere Aufteilungen[100]:
Bei kleinerer Darstellung von Abteilungen wird nur noch der Name dargestellt. Wird die Darstellung zu klein, so enthält sie keine Informationen mehr. Beim Überstreichen mit dem Mauszeiger zeigt dann ein graphischer Ballon Name, Abteilungsleiter und Größe.

3.3.2.1.2 Arbeitsgruppen

Arbeitsgruppen haben andere Aufgaben und Zusammensetzungen als Abteilungen. Ihre Mitglieder besitzen meist unterschiedliche Fähigkeiten und Merkmale. Ihr Fachwissen und Können soll sie zu einem ‘Team’ zusammenführen, in dem der Gruppengedanke im Vordergrund steht. Solche ‘Teams’ müssen sich allein schon optisch von Abteilungen unterscheiden.

Weiterhin haben Arbeitsgruppen keine Hierarchie. Strukturelle Verknüpfungen beziehen sich auf gleichrangige Arbeitsgruppen, die einander in einer oder doppelter Richtung zuarbeiten. Da alle Arbeitsgruppen hierarchisch gleichwertig sein sollen, sind sie am besten auf einer Ebene darzustellen. Da aber der Bildschirm eine senkrechte Ebene ist, würde eine zweidimensionale (also flache) Darstellung von Arbeitsgruppen eine Hierarchie implizieren. Aus dieser Vorüberlegung ist die Idee zu einer dreidimensionalen Darstellung von Arbeitsgruppen und ihrer Relationen entstanden, die optisch auf einer in den Bildschirm verlaufenden Ebene liegen sollen.

[100] vgl. Kap. 4.4.2.1.3 ‘Formatierfunktionen’

Dieser Effekt kann an folgendem optischen Beispiel nachvollzogen werden: Auf dem linken Bildschirm sind die oberen zwei Objekte den anderen anscheinend übergeordnet. Kippt man die optische Ebene etwas horizontaler (Bildschirm 2), wird der Eindruck vermittelt, die Objekte seien gleichrangig, da sie scheinbar auf gleicher Höhe liegen.

Abbildung 3-13 Zwei- und dreidimensionale Perspektive

Innere Aufteilung:

Eine Arbeitsgruppe verfügt über folgende Attribute[101]:
- Beschreibung,
- „Ist Arbeitsgruppen zugeordnet“,
- Name,
- Mitglieder,
- Anzahl der Mitglieder,
- Manager (evtl. nicht bestimmt).

Für die innere Struktur eignen sich nur die letzten vier Attribute, da Zuordnungen aus der Darstellung ersichtlich sind und eine Beschreibung zuviel Platz beanspruche. Um zu suggerieren, daß eine Arbeitsgruppe nicht vorrangig organisatorische Einheit ist, sondern seine Stärke durch die Gruppendynamik seiner Mitglieder entfaltet, werden die Mitglieder symbolisch dargestellt.

Die Größe einer Arbeitsgruppe wächst mit der Anzahl der Mitglieder. Damit besonders viele Mitglieder darstellbar sind, werden sie auf einer Ellipse am äußeren Rand angeordnet. Falls ein Manager existiert, wird dieser in der Mitte hinten dargestellt. Ansonsten bleibt diese Stelle leer. Für jede Arbeitsgruppe kann die ‘Animation’ auch ausgeschaltet werden. Sie wird dann in kleiner Größe ohne Personensymbole dargestellt.

[101] vgl. Entity-Relationship Modell in Kap. 3.2.1.3

Abbildung 3-14 Mehrere Arbeitsgruppen

In der prototypischen Umsetzung des Organisationsmodells haben auch Arbeitsgruppen abhängig von der Größe der Darstellung[102] verschiedene innere Aufteilungen:
Bei kleiner werdender Darstellung verschwinden die Personenobjekte. Später wird nur noch der Name dargestellt. Wird die Darstellung zu klein, so hat sie keine Informationen mehr im Inneren. Beim Überstreichen mit dem Mauszeiger zeigt ein graphischer Ballon Name, Manager und Größe.

Da die Personensymbole alle gleich aussehen, fehlt ein Unterscheidungsmerkmal zur eindeutigen Zuordnung auf eine Person. Diese Möglichkeit wird später interaktiv gegeben, indem ein Ballon mit dem Namen der Person eingeblendet wird, sobald der Mauszeiger die Person überstreicht. Weitere Interaktionsmöglichkeiten bestehen im einfachen Versetzen oder Kopieren eines oder mehrerer Personensymbole[103].

3.3.2.1.3 Personen und Rollen

Für Personen und Rollen existiert keine eigenständige relationale Struktur. Personen werden nur durch ihre Zugehörigkeit zu Abteilungen oder Arbeitsgruppen zueinander in Beziehung gesetzt. Deshalb wird keine graphische Darstellung einer speziellen Person benötigt. Rollen sind nur Eigenschaften einer Person und werden demzufolge nicht graphisch dargestellt.

---

[102] vgl. Kap. 4.4.2.1.3 ‘Formatierfunktionen’
[103] vgl. Kap. 4.4.2.1.2

Personensymbole in den Arbeitsgruppen repräsentieren keine komplette Person als Stelle, sondern zeigen nur, daß diese Person dort vertreten ist. So kann dieselbe Person durch mehrere Personenobjekte vertreten sein (in folgender Abbildung in gleicher Farbe dargestellt) und damit zu mehreren Arbeitsgruppen gehören:

Abbildung 3-15 Personen in mehreren Abteilungen

Da es meist mehr Personen gibt als verfügbare Farbnuancen, wurde bei der Bildschirmdarstellung auf eine unterschiedliche Farbgebung der Personenobjekte verzichtet. Eine andere Färbung der evtl. existierenden Arbeitsgruppenleiter wird nicht vorgenommen, da sie meist nicht aufgrund autoritärer Qualitäten bestimmt werden. Gleiche Farbe soll ihre Zugehörigkeit zur Gruppe unterstreichen.

3.3.2.2 Verknüpfende Darstellungen

Die Darstellung von Personenobjekten in oder optisch ‘auf’ den Arbeitsgruppen zeigt schon eine Verknüpfung von Objekten. Da die Anordnung der Personenobjekte einen großen Einfluß auf die dreidimensionale Perspektive einzelner Arbeitsgruppenobjekte nimmt, wurde diese Beziehung schon bei den Arbeitsgruppen erläutert.[104]

Weitere wichtige Beziehungen bestehen zwischen Abteilungen in der Abteilungshierarchie und zwischen Arbeitsgruppen.

3.3.2.2.1 Abteilungshierarchien

Für die Darstellung von Abteilungshierarchien in sog. **Organigrammen** gibt es in der Literatur eine Vielzahl von Vorschlägen. Blum beschreibt im wesentlichen folgende Formen:[105]

[104] vgl. Kap. 3.3.2.1.2
[105] vgl. **[Blum1980]** S. 46-91

Abbildung 3-16 Organigrammformen nach Blum

Die Abteilungshierarchie in der Aufbauorganisationsdatenbank ist ein Einlinienmodell. Als gebräuchlichste Form wird eine vertikale Pyramide verwendet:

Abbildung 3-17: Horizontale Pyramide

Auf diese Organigrammform kann nicht verzichtet werden. Sie bringt aber in der Bildschirmdarstellung einen Nachteil:
Wenn eine Darstellung größer als der Bildschirm ist, zeigt der Bildschirm nur einen Ausschnitt eines größeren sog. ‘Arbeitsblattes’[106]. Standardmäßig zeigt der Bildschirmausschnitt die linke obere Ecke des ‘Arbeitsblattes’. Der logische Einstiegspunkt für ein Organigramm wäre aber die oberste Abteilung, also die Firmenleitung. Diese Abteilung liegt meist in der Nähe der Mitte des Organigrammes. Bei großen Organigrammen ist sie daher anfangs nicht zu sehen, und der Bildschirmausschnitt muß erst verschoben werden.

Deshalb wird in der graphischen Darstellung zusätzlich eine Abwandlung der von Weidner vorgeschlagenen horizontalen Pyramide verwendet[107]. Die Abbildung 3-18 zeigt

[106] ein engl. Fachbegriff bezeichnet dies als ‘virtual screen’, also ‘virtuellen Bildschirm’
[107] vgl. **[Weidner1977]** S.146

links die horizontale Pyramide von Weidner und rechts eine Form, in der die Hauptabteilung bildschirmkonform immer links oben steht:

Abbildung 3-18 Vertikale Pyramiden

Die rechte Form zeichnet sich durch einen geringeren Platzbedarf aus. Andere Darstellungsformen für Einlinienmodelle sind nicht universell einsetzbar, sie scheiden lt. Blum wegen zu hohem Platzbedarf aus, oder weil sie nur kleinere Organisationen übersichtlich darstellen können.[108]

Zum Abschluß muß noch die Positionierung von Stabstellen bedacht werden. Im Rahmen des Organisationsmodells dieser Arbeit wird eine Stabstelle direkt einer Abteilung zugeordnet. Deshalb muß sie in unmittelbarer Nähe zu dieser Abteilung stehen. Eine Möglichkeit wäre, sie links oder rechts auf gleicher Höhe anzuordnen. Wenn aber mehr als zwei Stabstellen angegliedert werden, wird diese Darstellung zu breit. Deshalb werden die Stabstellen einer Abteilung unter ihr gezeichnet, jedoch über den normalen Unterabteilungen. Damit die Abteilungen einer Hierarchieebene weiterhin auf einer Höhe liegen, müssen sie entsprechend abgesenkt werden:

Abbildung 3-19 Organigramm mit Stabstellen

[108] vgl. **[Blum1980]** S. 46-51

3.3.2.2.2 Arbeitsgruppen ‘Netzwerke’

Relationen von Arbeitsgruppen haben einen informelleren Charakter. So ist unter der Zielsetzung der Vermeidung von Hierarchie keine Arbeitsgruppe einer anderen untergeordnet. Relationen besagen, daß eine gleichberechtigte Arbeitsbeziehung zwischen den Arbeitsgruppen besteht, wobei in einer oder beiden Richtungen zugearbeitet wird. Unter diesen Gesichtspunkten können Relationen zwischen Arbeitsgruppen unter zwei Perspektiven graphisch dargestellt werden:

-als Kommunikationsbeziehungen unabhängiger Einheiten

-oder als eine Mengendarstellung, in der Arbeitsgruppen Teilbestandteil anderer Arbeitsgruppen sein können.

Die zweite Perspektive wird meist in sog. Venn-Diagrammen ausgedrückt:

Abbildung 3-20 Arbeitsgruppen im Venn-Diagramm[109]

In der Abbildung umfaßt die Arbeitsgruppe C[110] zwei andere Arbeitsgruppen, auf deren Arbeitsergebnissen sie aufbauen kann. Die Darstellung im Venn-Diagramm hat den Vorteil, daß eine Person nur an einer Stelle eingezeichnet ist. So gehören z.B. die beiden Personen im roten Bereich[111] gleichzeitig zu den Arbeitsgruppen A und B. Sie liegen weiterhin im graphischen Bereich der Arbeitsgruppe C, gehören also auch zu C. Mit Hilfe von Venn-Diagrammen können Arbeitsgruppen in ihrem Aufbau recht intuitiv dargestellt werden.

Leider ist es im Rahmen dieser Arbeit nicht gelungen, Arbeitsgruppenstrukturen universell in Venn-Diagramme umzusetzen. Einige Anzeichen sprechen dafür, daß das auch nicht möglich ist:

Einerseits können Venn-Diagramme nur für Mengen eingesetzt werden, zwischen denen relativ wenige Zusammenhänge bestehen. Viele Zusammenhänge werden mit wachsender Anzahl zuerst unübersichtlich, dann mit immer größerem Aufwand erstellbar. Es können auch Zusammenhänge entstehen, für die eine zweidimensionale Mengendarstellung nicht mehr ausreicht.

---

[109] identisch mit Abbildung 2-3

[110] Team C

[111] in der Mitte

Zu der Frage, ob ein Modell graphisch darstellbar ist, kommt noch ein weiterer Aspekt: es muß von einer Rechenmaschine in akzeptabler Zeit berechenbar sein. Das ist besonders wichtig, wenn ein Benutzer interaktiv mit dem Modell arbeiten soll. Der Computer kennt intern nur Zahlen und muß sich zu Bildern erst 'seine Zahlen machen' oder zur Darstellung von Bildern erst jeden Punkt[112] berechnen. Bildhafte Zusammenhänge, die ein menschlicher Betrachter 'auf einen Blick' sieht, können einen Computer stundenlang beschäftigen.

Um alle möglichen Konstellationen darstellen zu können, die der spätere praktische Einsatz mit sich bringen kann, mußte bei der Darstellung von Arbeitsgruppen deshalb auf eine 'zweitbeste' Lösung zurückgegriffen werden. Wie im vorhergehenden Kapitel bereits beschrieben, wurde eine angedeutet dreidimensionale Perspektive der Arbeitsgruppen gewählt. Diese Perspektive soll suggerieren, daß Arbeitsgruppen hierarchisch auf einer Ebene liegen. Durch Linien werden informelle oder kommunikative Verbindungen angedeutet. Die Leistungsfähigkeit oder Kapazität einer Arbeitsgruppe ist subjektiv von ihrer Größe abzuleiten.

Arbeitsgruppen können animiert oder normiert dargestellt werden.[113] In der animierten Form werden Mitglieder als Personenobjekte abgebildet und die Größe ist abhängig von der Mitgliederzahl.

Abbildung 3-21 a Arbeitsgruppen in animierter Darstellung

[112] engl.: Pixel
[113] vgl. Kap. 3.4.1

Abbildung 3-21b: Arbeitsgruppen in teilweise animierter Darstellung

3.4 Exkurs: Sozialverträgliche Konzeption des Organisationsaufbaus mit Hilfe der Informationstechnologie

Ein Aufbauorganisationsmodell und damit die Gestaltung einer Aufbauorganisation greifen in vielfältigen Formen in die Zusammenarbeit der Menschen ein. „Noch ungenügend analysiert sind die bisher feststellbaren Wirkungen (und erst recht die künftigen Möglichkeiten) im Bereich der sozialen Formen, die einerseits durch die Implementierung von Kooperationsregeln in CSCW - Systemen entstehen, und andererseits durch die neuen Möglichkeiten der Leistungskontrolle, der verschärften Arbeitsteilung, aber auch der Enthierarchisierung von Organisationen [...].[114]“

„War bisher schon die große Bedeutung der Kommunikation für Büro und Verwaltung deutlich, so zieht mit der CSCW - Technologie die Technik ins Büro ein. Von Anfang an war die Telekommunikation mit beteiligt und gestattet fast beliebige Mensch-Maschine-Kommunikation über bisherige Bürogrenzen hinaus, die zusätzlicher Planung und Entscheidung bedarf, um benutzerfreundlich zu werden. Daraus resultieren zwei Folgen für die Ablauf- und Aufbauorganisation: Erstmals seit Beginn der industriellen Arbeitsteilung werden Teilfunktionen (auch verschiedene Personen, Berufe, Behörden, Unternehmen ...) wieder zusammenführbar (‘integrierbar’). Vor allem werden erst Arbeitsplätze, dann ganze Büros untereinander in den unterschiedlichsten Formen und ‘Architekturen’ vernetzt:[115] „Die alten Einzelbüros werden zu höheren Organisations-

[114] **[Raeithel1991]** S.139
[115] **[Steinmüller1993]**S.313, vgl. auch Kap. 2.1.3

formen koordiniert und zusammengeschaltet ('integrierte Bürokommunikation'), auch um den Vorgesetzten unter den gewandelten Bedingungen mehr Einfluß zu geben.“[116]

Ein System, das solch tiefgreifende Änderungen mit sich bringt und mit der sozialen Eigendynamik seiner Mitglieder operiert, muß auch kritisch hinterfragen, ob diese weitreichenden Manipulationen akzeptierbar sind. Es muß abwägen, inwieweit Persönlichkeitsstrukturen und soziale Beziehungen seiner Mitarbeiter berührt werden und schützenswert sind. Das begründet sich zum einen aus moralischer Verantwortung, zum anderen auch aus einem gewandelten öffentlichen Bewußtsein, dem jedes Unternehmen Rechnung tragen sollte. Wie in der Einleitung schon erwähnt, treten in den heutigen Unternehmensformen andersartige Eigenschaften bzw. Fähigkeiten als bisher in den Vordergrund. Das sind z.B. Kreativität, Flexibilität, hohes Informationsverarbeitungsvermögen, rationale Konfliktaustragung, Innovationsfähigkeit und höheres differenziertes Selbstregulationspotential. Neben der produktiven Rechtfertigung der Unternehmensführung und der Unternehmensberater auch deshalb, weil der Mensch im Zuge der Humanisierung der Arbeitswelt nicht mehr als Produktionsfaktor angesehen wird und angesehen werden darf. Um seine Kreativität nutzen zu können, muß er einen höheren Informationsstand erhalten und zwangsläufig einen höheren Standard an Verantwortung.

Welche Bewertungskriterien gibt es, an denen man messen kann, ob eine neue Technologie sozialverträglich ist?

Soziale oder sozialökonomische Analyse untersucht den Standpunkt des allgemeinen Nutzens. Sie zielt primär auf die „Nützlichkeit für die Menschheits-Gesellschaft, vor allem für die unmittelbar oder mittelbar Betroffenen, erst nachgeordnet auf die Realisierung sonstiger partikulärer Interessen, etwa eines Betreibers.“[117]

Mögliche Bewertungskriterien eines CSCW - Systems und damit auch der Aufbauorganisation sind beispielsweise:
- Ist ein System rechtlich zulässig?
- Wird es vom Benutzer akzeptiert?
- Vernichtet es Arbeitsplätze?
- Macht es zufriedener?
allgemein: welchen gesellschaftlichen Nutzen oder Schaden stiftet es?

Der Aspekt rechtlicher Zulässigkeit soll an dieser Stelle nicht bearbeitet werden, stellt aber trotzdem ein gewichtiges Kriterium dar. Es darf beim praktischen Einsatz von CSCW - Systemen und der daraus resultierenden Datenhaltung nicht außer acht gelassen werden.

Anhand obiger Kriterien läßt sich recht schnell eine bewertende Kritik begründen. Soziale Bewertung und Analyse sollte jedoch nicht bei negativer Kritik steckenbleiben,

[116] ebenda S. 313 f.
[117] ebenda S. 265

sondern *konstruktive* Veränderung anstreben. Daher sollen ein paar direkte Kritikpunkte zur Diskussion gestellt werden:

Unbestreitbar ist, daß effizientere organisatorische Strukturen den Einzelnen leistungsfähiger machen. Ökonomisch gesehen ist aber eine Arbeitsentlastung, wie sie gerne im Bereich der PC - gestützten Automation propagiert wird, unsinnig. Entweder soll die gleiche Anzahl an Menschen mehr leisten, oder die bisherige Arbeitsmenge wird kostengünstiger auf weniger Menschen verteilt. Deshalb wird die Arbeit eines einzelnen Menschen nicht einfacher. Eine Aufteilung der anfallenden Arbeit auf weniger Menschen ist mit Freisetzungen verbunden, die durch Entwicklerjobs im CSCW - Bereich nicht aufgewogen werden können. Allerdings ist das gewissermaßen eine philosophische Frage, da bei Nichtkonkurrenzfähigkeit der Unternehmung diese Menschen evtl. sowieso ihre Arbeit verlieren würden.

Dafür spricht auch folgender Aspekt: Eine effizientere Aufgabenerfüllung durch flexiblere Organisationsformen wird recht schnell von Konkurrenten adaptiert werden. Daraus resultiert wieder ein Gleichstand der Voraussetzungen. Um dann noch Vorteile zu realisieren, wird eine verbesserte, flexiblere Organisation in den meisten Fällen die Aufgabenkomplexität erhöhen, was für die Arbeitnehmer keine Entlastung mit sich bringt, sondern ein erhöhtes Anforderungspotential.[118]

Die Entwicklung darf nicht einseitig negativ gesehen werden. Es ist möglich und wünschenswert, daß Strukturformen eine Gruppendynamik entwickeln, mit der die Mitglieder bei höherer Motivation bessere Ergebnisse erzielen. Wahrscheinlich kann eine partizipative Gestaltung der organisatorischen Strukturen hemmende soziale Faktoren beseitigen. So spricht Frese bürokratischen Strukturen „dysfunktionale Folgen“ zu: „Bürokratische Strukturen [...] [führen] bei den Mitarbeitern zu Frustationen, Versagen, Verfolgung, lediglich einer kurzen Perspektive und zu Konflikten.“[119] Er erkennt vor diesem Hintergrund, daß unbürokratische Strukturen solche Folgen mildern können: „Eingehende Untersuchungen über den Einfluß unterschiedlicher Organisationsstrukturen auf die Fähigkeit eines organisatorischen Systems, Innovationen hervorzubringen, sind vor allem in der Kleingruppenforschung und für den Forschungsbereich industrieller Unternehmen durchgeführt worden [...]. Die Ergebnisse sind - bezogen auf diese Bereiche - relativ einfach: Sie betonen die Vorteilhaftigkeit möglichst ungebundener Strukturen, insbesondere den Abbau von Kommunikationsbeschränkungen.“[120]

Mit den Kritikpunkten vor Augen könnte man sagen, daß eine Aufbauorganisationsdatenbank nur einen sehr kleinen Bestandteil der neuen Technologien im Bereich der Bürokommunikation und -automation darstellt. Sie bietet aber in ihrer Konzeption den wesentlichen Ansatzpunkt der Partizipation. Sie dient als leicht bedienbares Strukturierungswerkzeug, mit dem alle Beteiligten an der Datenmenge, also dem Wissen des

---

[118] vgl. **[WainFran1984]** S. 36
[119] **[Frese1987]** S. 297
[120] **[Frese1987]** S. 298

Unternehmens, teilhaben können. Darüber hinaus ermöglicht sie eigenverantwortliche Mitarbeit.

Ein Nebeneffekt partizipativer Gestaltung der Aufbauorganisation ist, daß der Datenbestand für alle transparent gemacht wird. Wilhelm Steinmüller beurteilt Daten als Machtfaktoren: „Daten sind Machtfaktoren. Die Bezüge zum Datenherrn und Datenobjekt sind Herrschaftsbezüge“[121]. Eine partizipative Systemgestaltung und gemeinsame Nutzung nimmt Datenbeständen ihren (un-)heimlichen Charakter. Wissen kann seine Macht nur solange ausüben, wie es einzelnen vorenthalten wird. Wie Arne Raeithel sagt, wird die „Abgrenzung und Auswahl der Mittel [...] normalerweise durch die faktischen Machtverhältnisse zwischen den kooperierenden Personen aufrechterhalten.“[122] Je gleichberechtigter eine Kooperationsform ihre Mitglieder behandelt, desto sozialverträglicher sind ihre Auswirkungen.

Aus diesem Grunde ist der Versuch zum Abbau von Hierarchie auch aus sozialkritischer Sicht positiv zu bewerten. „Gemeinschaftliches Handeln ist [...] erreichbar, wenn jeder Akteur seinen Anteil an den gemeinsamen Handlungen präzise an der gemeinsamen Mittel-, Ziel- bzw. Kooperationsstruktur ausrichtet. Dazu ist es aber weiterhin erforderlich, daß er andererseits möglichst genau die Beiträge der anderen antizipieren kann, um wiederum seine Handlungen [...] an die der anderen anschließen bzw. anpassen zu können.“[123] Je genauer eine Person die Ziele, Kooperationsmuster und Beiträge anderer Personen kennt, desto weniger muß sie sich auf die formale Hierarchie berufen und um so besser werden heutige Forderungen aus dem sozialorientierten Bereich verwirklicht werden können.
Dazu gehören z.B.:

- Delegation der Entscheidung und damit Entlastung der Leitung,
- Reduzierung der Komplexität organisatorischer Gestaltungsentscheidungen,
- Verbesserung der lateralen[124] Kommunikation ohne die Zwänge und Ängste der Rangordnungen und deren taktischen Manöver zur Besitzstandswahrung.

Im Rahmen dieser Ausführungen, die die Problematik nur oberflächlich streifen konnten, kann man sich Arne Raeithel anschließen:
„Wie bei allen Maschinensystemen ist eine Rückwirkung der neuen Arbeitsplätze auf die Tätigkeits- und auch Persönlichkeitsstruktur der Arbeitenden unvermeidlich. ... [Wahrscheinlich] sind die Chancen für humanere Arbeitsgestaltung hierbei größer als die Gefahren, aber dies muß sich erst noch erweisen.“[125]

---

[121] **[Steinmüller1993]**S. 351
[122] **[Raeithel1991]**S. 136
[123] **[Pasch1991]**S. 118
[124] nicht hierarchischen
[125] **[Raeithel1990]**S. 132

Kapitel 4: Gestaltung der Implementation

Bisher wurde ein theoretisches *Metamodell* zur Entwicklung und Verwaltung von Aufbauorganisationen geschaffen. Es enthält konkrete organisatorische Einheiten und deren Beziehungen, ein universelles Organisationsmodell für Abteilungen und eine Verknüpfungsform für Arbeitsgruppen. Zur Repräsentation der Aufbauorganisationen sind graphische Modelle diskutiert und entwickelt worden, die sich besonders für die Darstellung am Bildschirm eignen. Das Metamodell erlaubt somit die Abbildung organisatorischer Strukturen sowohl in Datensätzen einer Datenbank, als auch in graphischer Darstellung, die für den menschlichen Benutzer verständlicher ist.

Dieses Kapitel beschreibt die praktische Umsetzung des Metamodells in einer PC - gestützten Entwicklerumgebung, die konkrete Aufbauorganisationen modellieren und verwalten kann. Dazu werden organisatorische Daten eines Unternehmens mit Hilfe eines graphischen Modellierungswerkzeuges, dem ‘Organization-Object Modeler’ erzeugt, in einer Notes - Datenbank abgelegt und verwaltet.

Zu einer kompletten (Organisations-) Datenbank gehören der Definition aus Kapitel 2.1.2 zufolge eine Datenbasis und ein Modul, das zur Verwaltung und als Schnittstelle für andere Applikationen dient. Die Datenbasis einer Aufbauorganisation wird in einer Lotus - Notes Datenbank abgelegt und mit Notes verwaltet. Da die Applikation ‘Notes’ als Schnittstelle über keine geeigneten graphischen Fähigkeiten und keine passenden tabellarischen Bearbeitungsmöglichkeiten verfügt, ist sie zur Entwicklung und effizienten Bearbeitung einer Aufbauorganisation wenig geeignet. Sie wird durch den Organization-Object Modeler als Werkzeug zur graphischen Visualisierung ergänzt. Er unterlegt graphisch dargestellte Objekte mit eigener Funktionalität und unterstützt somit interaktives In-Beziehung-Setzen der Datenstrukturen. Durch schnelle graphische Zuordnung werden Relationen erstellt, die in textorientierten Datenmasken bedeutend aufwendiger zu realisieren wären. Damit dient der Organization-Object Modeler als eine Art Übersetzer, der sequentielle Strukturen der Datensätze in objektorientierte, graphische Formen aufbereitet. Nach der Bearbeitung durch den menschlichen Benutzer formt er die graphischen Darstellungen zurück und sichert sie in der Notes - Umgebung. Damit übernimmt er die gesamte Bearbeitung der Aufbauorganisation, so daß der menschliche Benutzer die Notes - Umgebung nicht mehr direkt benutzen muß. Sie wird nur noch als Hintergrundapplikation vom Organization-Object Modeler gestartet.
Aufgrund dieses Zusammenhanges, unterstützt der Organization-Object Modeler alle unter Notes üblichen Zugriffsbeschränkungen. Deshalb werden bei der Arbeit mit einigen Datenbanken Paßworte benötigt, die abgestufte Rechte freigeben oder versagen.

Bei der Konzeption der Applikation wurde besonderer Wert auf leicht verständliche Darstellungen gelegt, damit sie für alle Mitarbeiter einer Büroumgebung nutzbar wird,

und sie damit den Forderungen nach eigenverantwortlicher, partizipativer Gestaltung und flexibler Veränderung der Aufbauorganisation entspricht.[126]

Innerhalb dieses Kapitels wird zunächst der allgemeine Aufbau der graphischen Benutzerschnittstelle vorgestellt. Darauf folgt eine kurze Beschreibung der Datenstrukturen, die die Applikation verwendet, sowie der für Lade- und Speichervorgänge nötigen Umformungen. Ein dritter Teil behandelt systematisch das Fenstersystem, mit dessen Hilfe eine Aufbauorganisation innerhalb des Organization-Object Modelers dargestellt und bearbeitet wird. Zum Ende des Kapitels folgt eine Beschreibung besonderer Funktionalitäten, die neben der einfachen *Präsentation* einen komfortablen und rationellen *Umgang* mit den graphischen Strukturen ermöglichen sollen.

Ein kurzer Ausblick beschäftigt sich mit Erweiterungsmöglichkeiten und beurteilt bekannte Schwächen der Applikation.

4.1 Der Prototyp „Organization-Object Modeler“

Der ‘Organization-Object Modeler’ ist eine universelle Applikation zur Darstellung und Bearbeitung von organisatorischen Datenbanken. Wie in der Einleitung beschrieben, wurde er für die graphischen Benutzeroberflächen der Betriebssysteme MS-Windows und OS/2 implementiert. Er paßt sich den Gegebenheiten des jeweiligen Betriebssystems und der Benutzeroberfläche an, so daß es geringe Unterschiede im äußeren Erscheinungsbild und in funktionalen Details geben kann. Die Gestaltung der Benutzerschnittstelle entspricht den üblichen Gestaltungsrichtlinien der Betriebssystemoberflächen **MS-Windows** und des **Presentation Managers** unter OS/2 und bietet damit allgemein übliche Standardfunktionalitäten. Diese Standardelemente werden im folgenden Kapitel näher beschrieben.[127]

Die Applikation kann beliebig viele Datenbanken gleichzeitig öffnen. Sie gestattet auf einfache Weise Netzwerkzugriffe auf externe Server und kann damit auch beliebige externe Datenbanken bearbeiten.
Der Organization-Object Modeler stellt Interaktionsmöglichkeiten zur Verfügung, mit denen auch *zwischen* Datenbanken beliebige Objekte und Strukturen ausgetauscht werden können. Damit wird der Grundstein für den Aufbau von Organisationsbibliotheken gelegt, weil einmal erstellte organisatorische Strukturen in beliebiger Konstellation wiederverwendet und erweitert werden können.[128] Durch nahezu beliebiges Verknüpfen von graphischen Objekten entsteht auch *innerhalb* einzelner Datenbanken eine Vielzahl möglicher Interaktionsvorgänge, deren Bedeutung durch kontextsensitive Mauszeiger[129] plastisch unterschieden wird. Für graphische Darstellungen stehen mächtige Formatier-

[126] vgl. Kap. 3.1.1
[127] vgl. Kap. 4.1.1
[128] vgl. Kap. 4.4.1
[129] engl.: pointer

funktionen zur Verfügung, die Anpassungen an den benutzten Bildschirm, aber auch unterschiedliche Perspektiven auf dargestellte Strukturen ermöglichen.[130]

Bei der Implementation wurde Wert auf konsequenten objektorientierten Aufbau gelegt, so daß alle graphischen Elemente mit eigener Funktionalität ausgestattet sind. Daraus soll sich ein Zusammenspiel ergeben, das reale aufbauorganisatorische Strukturen möglichst detailgetreu wiedergibt und dem menschlichen Bearbeiter formale Regelungen der Aufbauorganisation plastisch durch das Verhalten der Objekte suggeriert.

Jedes graphische Element verfügt über ein eigenes Kontextmenü, das sich auf Klicken der rechten Maustaste öffnet. Darin können lokale Funktionen aufgerufen oder ein Dialogfenster mit Eigenschaften des Objektes geöffnet werden

Die Applikation zeichnet sich durch einfache Installation aus, bei der alle erforderlichen Dateien in einem Verzeichnis abgelegt werden, ohne daß Einträge in Systempfaden oder sonstige ini - Dateien in anderen Verzeichnissen erforderlich sind. Desweiteren haben sich in einigen Testläufen hohe kapazitative Fähigkeiten gezeigt, so daß die Applikation auch große Datenbanken laden und präsentieren kann.[131]

Bemerkung:
Eine Einbuße im Komfort der Bearbeitung ergibt sich jedoch bei sehr großen Datenbanken, weil kontextsensitive Abgleiche, der Größe entsprechend, etwas langsamer werden. Jedoch sind extreme Größenordnungen im Umfeld herkömmlicher Büroumgebungen offensichtlich nicht zu erwarten.[132]

Neben der Entwicklung für verschiedene Betriebssysteme[133] sind jeweils englische und deutsche Versionen entstanden. Eine spanische Version ist vorbereitet, aber nicht vollständig übersetzt.

4.1.1 Standardelemente der graphischen Benutzeroberfläche (GUI)

Um den gängigen Standards heutiger graphischer Benutzeroberflächen zu entsprechen, sind im Organization-Object Modeler folgende funktionalen Elemente implementiert worden:

- Ein MDI - Interface mit verschiedenen, kontextabhängigen Menüs,
- eine Werkzeugleiste (engl.: Toolbox),
- eine Statusleiste (engl.: Statusbar),
- eine Zwischenablage für alle verwendeten Datenformate,
- ein Hilfesystem,

[130] vgl. Kap. 4.3.1.1.3
[131] vgl. Kap. 4.5
[132] vgl. ebenda
[133] engl.: ‘Cross Platform Development’

- Funktionen zur Druckvorschau und zum Drucken.

Die Ausgestaltung der beschriebenen Elemente folgt im wesentlichen dem IBM SAA/CUA Standard,[134] der den Aufbau von Applikationen unter OS/2 oder MS-Windows beschreibt. Die letzten beiden Funktionen bilden in sich abgeschlossene Themen mit denen sich zwei Unterkapitel befassen.[135] Die anderen Elemente werden an dieser Stelle näher beschrieben. Neben der technischen Erläuterung enthalten einige der Ausführungen Verweise auf weiterführende Kapitel, die sich mit spezielleren Aspekten und Nutzungsmöglichkeiten beschäftigen.

Zum Überblick ist in folgender Abbildung die **deutsche** Version des Organization-Object Modeler unter **Windows95** zu sehen. Die aus Kapitel 3 bekannten gaphischen Darstellungen sind um 50% (bei der Abteilungshierarchie) und um 15% (bei den Arbeitsgruppen) verkleinert.[136]

Abbildung 4-1: Der Organization-Object Modeler

Zusätzlich zu den graphischen Strukturen enthält der Organization-Object Modeler verschiedene Tabellendarstellungen, sogenannte 'Browser'.[137] In der Abbildung ist der Personen-Browser zu sehen, eine Tabelle aller Personen einer Datenbank.

---

[134] vgl. **[IBM1991]**
[135] vgl. Kap. 4.1.1.1 und Kap. 4.1.1.2
[136] vgl. Kap. 4.3.1.1.3 'Formatierfunktionen'
[137] vgl. 4.3.1.2

Die MDI - Technik und verschiedene Menüs

Die Applikation präsentiert sich mit einem Applikationsfenster, in dem beliebig viele 'Dokumentfenster' erzeugt und angeordnet werden können. Diese MDI - Technik[138] erlaubt eine effiziente Darstellung von einzelnen Strukturen einer Datenbank, aber auch eine gleichzeitige Bearbeitung beliebig vieler Organisationsdatenbanken. Alle geöffneten Dokumentfenster[139] werden innerhalb des Applikationsmenüs eingetragen und können auf unterschiedliche Weise gegeneinander ausgerichtet oder aktiviert werden.
Ist kein MDI - Fenster geöffnet, zeigt die Applikation nur ein eingeschränktes Menü, in dem nicht alle Menüpunkte und -funktionen zu sehen sind. In beiden Menüs werden Menüpunkte kontextabhängig aktiviert - oder, wenn sie im Kontext keinen Sinn machen, „ausgegraut". Innerhalb der Menüs sind für verschiedene Befehle sogenannte **'Shortcuts'** definiert. Das sind Tastaturkombinationen, mit denen der entsprechende Menüpunkt aufgerufen werden kann. Die Shortcuts belegen je nach Art und Version des Betriebssystems unterschiedliche Tasten.

Zusätzlich zu den MDI - Fenstern benutzt der Organization-Object Modeler eine Vielzahl von **Dialogfenstern**, die in einigen Fällen die übrige Applikation sperren und zunächst bearbeitet werden müssen.[140] In Abbildung 4-1 ist rechts ein nicht-modales Dialogfenster zu sehen, das eine Zuweisung von Personen auf Arbeitsgruppen unterstützt.[141]

Die Werkzeugleiste

Unterhalb des Applikationsmenüs ist die Werkzeugleiste angeordnet. Sie enthält Sinnbilder für häufig verwendete Menübefehle, jeweils nach den entsprechenden Untermenüs gruppiert.

Bemerkung:
Zusätzlich zu den Menübefehlen wurde die Möglichkeit aufgenommen, Lotus Notes direkt auf Knopfdruck zu starten.[142] Dabei wird Notes automatisch gesucht, ohne daß entsprechende Verzeichnisse explizit angegeben werden müssen.

Die Statusleiste

Die Statusleiste steht am unteren Rand der Applikation und zeigt in den mittleren beiden Feldern Informationen über die aktuelle Datenbank. Das linke Feld dient für Statusmel-

---

[138] MDI = multiple document interface
[139] sog. 'MDI - Fenster'
[140] sog. modale Dialogfenster
[141] vgl. Kap. 4.4.4
[142] zweites Sinnbild von rechts

dungen der Applikation, wenn sie Berechnungen oder Datenabgleiche vornimmt. Im rechten Feld sind bei graphischen Darstellungen Koordinaten zur Navigation zu sehen.

Die Werkzeugleiste und die Statusleiste können bei Bedarf ausgeblendet werden, um die verfügbare Fläche im Applikationsfenster zu vergrößern.

Die Zwischenablage

Als letztes der wesentlichen Standardelemente ist eine applikationsinterne Zwischenablage vorhanden. Sie kann einzelne Objekte oder auch komplette Strukturen ausschneiden oder kopieren, um sie an anderen Stellen wieder einzufügen. Bei der Konzeption der Zwischenablage konnte nicht auf Standardelemente zurückgegriffen werden, da Objekte oder Strukturen einer Aufbauorganisation keine einfachen Datenstrukturen darstellen, sondern jeweils eine komplexe Anzahl von Attributen besitzen. Die Zwischenablage ist daher zu anderen Applikationen nicht kompatibel.

Bemerkung:
Die Zwischenablage ist das mächtigste Werkzeug zur generischen Wiederverwendung einmal erstellter Strukturen, da sie den Austausch von Objekten und auch von Objektstrukturen zwischen mehreren Datenbanken ermöglicht. Eine ähnliche aber eingeschränkte Funktionalität, bietet das direkte Kopieren oder Verschieben von Objekten mit der Maus. Dabei werden jedoch keine Relationen übertragen, sondern die Elemente gleichrangig eingefügt. Beide Funktionalitäten ermöglichen die Verwendung von Bibliotheken und werden in diesem Zusammenhang in Kapitel 4.4.1 weitergehend erläutert.

4.1.1.1 Drucken und Druckvorschau

Alle Graphiken können in einer Druckvorschau angesehen und auf Papier ausgedruckt werden. Es wurden keine Druckfunktionen für Tabellenstrukturen[143] implementiert, da dafür schon Funktionalitäten unter Lotus Notes verfügbar sind.

Zum Ausdrucken gibt es drei Funktionen:
Die Funktion **‘Druckereinrichtung’** sucht den aktuell gültigen Drucker und ruft dessen Einstellungsparameter auf. **‘Druckvorschau’** zeigt seitenweise das spätere Druckbild auf dem Bildschirm. Es können eine oder zwei Seiten nebeneinander dargestellt und gezoomt werden. Zoomen in der Druckvorschau hat keine Auswirkung auf den Ausdruck. Die Funktion **‘Drucken’** startet einen Ausdruck und enthält gewohnte Parameter für zu druckende Seiten und zur Qualitätsangabe.[144]

[143] vgl. Kap. 4.4.2.2
[144] z.B. Hoch- oder Querformat, Druckdichte usw.

Diese Möglichkeiten des Druckens und eine Druckvorschau sind in den Augen des Benutzers Standardelemente. Zum Bereitstellen der Funktionen im Organization-Object Modeler sind einige Kompromisse eingegangen worden, da organisatorische Strukturen nicht wie Texte direkt in den zweidimensionalen Begrenzungen von Papierformaten wiedergegeben werden können.

Ein Organigramm wird am Bildschirm auf mindestens einem, beliebig großen Arbeitsfeld modelliert. Ein Arbeitsfeld entspricht dem Inhalt eines MDI-Fensters. Um dieses Arbeitsfeld auf das Medium „Papier“ zu übertragen, wird es gerastert und in Stücken ausgedruckt. In den meisten Fällen paßt ein Arbeitsfeld auf **ein querliegendes** Blatt. Mehrere Blätter müßten daher zusammengelegt werden, um ein Arbeitsfeld komplett zu überblicken.
Neben dieser Rasteranordnung kann der Inhalt eines Arbeitsblattes gezoomt werden. Dabei übernehmen die Module für Drucken und Druckvorschau den Zoom-Faktor des Arbeitsblattes. Um einen Ausdruck auf ein Blatt zu begrenzen, könnte also wie folgt vorgegangen werden:
In der Druckereinrichtung stellt man den Ausdruck analog zum Bildschirmformat auf querliegendes Papier ein. Die Funktion ‘Druckvorschau’ informiert dann über Anzahl und Aussehen der benötigten Seiten. Kommt mehr als eine Seite zustande, kann der Inhalt des Arbeitsblattes durch Zoomen im MDI-Fenster verkleinert werden. Danach zeigt wiederum die Druckvorschau, wieviele Blätter ausreichen.

Bei jedem Ausdruck kann wahlweise ein Titelblatt mit Informationen über die dargestellte Graphik ausgegeben werden.

4.1.1.2 Das Hilfesystem

Ein Hilfesystem ist dazu da, den Benutzer mit Informationen über eine Applikation oder über bestimmte Teile einer Applikation zu versorgen. Damit sollen ihm z.B. Hilfen zur Menüauswahl, zu einzelnen Funktionen[145] oder zu bestimmten Schlüsselwörtern gegeben werden.
Im Organization-Object Modeler sind drei Arten von Hilfefunktionen vorgesehen.

- Fenster, Menüeinträge oder graphische Objekte bekommen einen Hilfetext zugeordnet, der bei bestimmten Aktionen[146] angezeigt wird.
  Zum Anzeigen werden meist kleine Ballons verwendet, die beim Überstreichen mit dem Mauszeiger auftauchen. Diese Funktion ist besonders beim neuen Umgang mit dem Organization-Object Modeler hilfreich, um die Bedeutung einzelner Menüpunkte oder Knöpfe zu erlernen. Später kann sie dann im Menüpunkt ‘?’ dauerhaft ausgeschaltet werden.

[145] z.B. Dialogfenstern
[146] z.B. Fokuserhalt eines Fensters, Aktivierung eines Menüpunktes, Überstreichen mit dem Mauszeiger

- Die zweite Hilfefunktion verwendet das jeweilige Hilfesystem des zugrundeliegenden graphischen Betriebssystems. Das ist unter OS/2 z.B. die ‘Information Presentation Facility’. Das Hilfesystem wird durch Betätigung der Hilfetaste des Betriebssystems[147] aktiviert. Für bestimmte Situationen sind Hilfekennziffern definiert, die zur Anzeige dafür vorgesehener Texte und Graphiken führen. Ist keine Hilfekennziffer aktuell, zeigen Hilfesysteme dieser Art eine generelle Einführung und Übersicht über das Produkt.

  Anmerkung:
  Die zweite Form der Hilfe ist im Organization-Object Modeler bisher nur vorbereitet und kann noch nicht genutzt werden. Es sind bereits Hilfekennziffern[148] im Quellcode definiert und an zentraler Stelle kurz erläutert. Zur Laufzeit der Applikation wird kontextsensitiv jeweils eine Hilfenummer aktuell gesetzt. Jedoch sind noch keine Hilfetexte formuliert worden, da die Kennziffern und Hilfetexte recht aufwendig für jedes Betriebssystem in einer speziellen Form erstellt werden müssen[149]. Eine entsprechend erstellte Hilfedatei kann in der Initialisierungsdatei ‘gf-orga.ini’ eingetragen werden. Dann wäre das Hilfesystem ohne Neuübersetzung nutzbar.

- Als drittes ist ein Feld in der Statuszeile am unteren Rand vorgesehen, um Arbeitsvorgänge anzuzeigen. Wenn die Applikation mit Berechnungen beschäftigt ist, so reagiert sie vorübergehend nicht auf Benutzereingaben. Dabei verwandelt sich der Mauszeiger in eine Uhr. Der Benutzer kann dann in allen Fällen die gerade durchgeführten Aktionen in der Statuszeile erkennen.

Eine allgemeine Hilfefunktion ‘Hilfe benutzen’ im Untermenü ‘?’ erläutert das betriebssystemspezifische Hilfesystem.
Zusätzlich zu den technischen Hilfen sind noch Funktionen zur statistischen Analyse einer Organisationsdatenbank gegeben. Sie werden in Kap. 4.5.4 näher erläutert.

4.1.2 Verwendete Entwicklungswerkzeuge

Der Organization-Object Modeler ist auf verschiedenen Betriebssystemen in der Sprache **C++** implementiert. Dafür wurde unter MS-Windows der Compiler **Visual C++ 1.5** und unter OS/2 der **IBM C Set 2 ++** Compiler verwendet. Die entsprechenden **Debugging**-Werkzeuge sind im Anhang unter Kapitel A-3 erläutert.

Die Aufbauorganisationsdatenbanken werden unter Lotus NOTES verwaltet. Notes bietet ein Interface zum Zugriff auf Datenbanken, das mit Hilfe von sog. API - Routinen[150] angesprochen werden kann. Zur Implementierung des Organization-Object

---

[147] unter OS/2 oder MS-Windows mit <F1>
[148] engl. ‘Hooks’
[149] vgl. **[StarView1992b]** Kap. ‘Hilfesysteme’
[150] API = ‘Advanced Programmer’s Interface’, Makrobefehle zum Datenbankzugriff

Modelers wurden die API Routinen nicht direkt benutzt, sondern eine Laufzeitbibliothek, die **Macroware - DLL,** verwendet. In der Macroware - Bibliothek sind NOTES APIs auf einem hohen Abstraktionsniveau zu Makrobefehlen zusammengefaßt. Mit Hilfe dieser Makrobefehle kann der Zugriff auf Datenbanken effizient mit kurzen Befehlsse-quenzen implementiert werden.[151]

Ein Vorteil der relativ allgemein gehaltenen Macroware-Befehle besteht darin, daß durch Anpassung der Makrobefehle oder Abfangen der Macroware-Anfragen mit Hilfe von Zwischenfiltern relativ einfach auch andere Datenbanksysteme als Speichermedium verwenden werden können.

Die Cross - Platform Entwicklung wurde durch die **StarView-Klassenbibliothek** unter-stützt. Diese Bibliothek bietet Klassen zur Implementierung der graphischen Benutzer-oberfläche. Ein mit ihnen erzeugter Quellcode wird auf den Zielbetriebssystemen über-setzt und soll ohne Adaptionen lauffähig sein. Dazu benutzt die fertige Applikation betriebssystemspezifische DLL - Dateien, die mit keinem Copyright versehen, also frei verfügbar sind.

Anmerkung
Bei der Portierung mit StarView treten einige Unzulänglichkeiten und zu beachtende Nebeneffekte auf. Diese werden z.B. in **[Meyer1995]**[152] erläutert. Trotzdem stellte sich die StarView - Umgebung bei der Implementation als ein leistungsfähiges Werkzeug heraus.

4.2 Datenhaltung und Datenabgleich mit der externen Organisationsdatenbank

Der Organization-Object Modeler muß zur interaktiven Bearbeitung alle Elemente einer Aufbauorganisation kennen. Deshalb liest er eine Organisationsdatenbank komplett ein und erzeugt gleichzeitig temporäre, interne Objekte, die die **Attribute** der Datensätze bekommen. Aus den Organisationsattributen berechnet er für die internen Objekte **Relationen**, die eine schnelle Zuordnung untereinander ermöglichen.

Damit ist die Datenhaltung innerhalb der Applikation aufwendiger als in der Notes - Datenbank. Dieses Kapitel gibt eine kurze Beschreibung der Datenstrukturen, die der Organization-Object Modeler zur Laufzeit erzeugt. Danach befaßt es sich mit den Funk-tionen zum Laden und Speichern einer Datenbank und beschreibt die Verfahren bei der Konvertierung in beide Richtungen.

---

[151] vgl.: **[Heindörfer 1993]**
[152] Kapitel 5

4.2.1 Applikationsinterne Datenstrukturen

In diesem Abschnitt soll verdeutlicht werden, welche Möglichkeiten eine objektorientierte Implementierung bietet. Eine detaillierte Beschreibung aller implementierter 'Objektklassen' kann jedoch nicht erfolgen, da der Quellcode fast nur aus eben diesen Klassen besteht und einen recht großen Umfang angenommen hat. Für den programmiertechnisch versierten Leser ist im Anhang eine Beschreibung aller Klassen aufgeführt, anhand derer die entsprechenden Schnittstellendefinitionen und -beschreibungen in den Quelldateien gefunden werden können.[153]

Bemerkung:
Bei der objektorientierten Implementierung spricht man von **Klassen**, wenn die Struktur bzw. der Typ eines 'Objektes' beschrieben wird. Beim Organization-Object Modeler entspricht ein Objekt beispielsweise einer organisatorischen Einheit. Ein konkretes Objekt ist eine **Instanz** seiner Klasse. Somit stammen mehrere gleichartige Objekte von derselben Klasse ab, z.B. sind alle Personenobjekte Instanzen einer Klasse 'Person'.

Der Organization-Object Modeler erzeugt für jedes Element der Aufbauorganisation ein eigenständiges Objekt. Dieses Objekt kann eine Person, eine Abteilung, eine Arbeitsgruppe oder Rolle sein. In jeder der vier Ausprägungen bekommt es alle Attribute des entsprechenden Datensatzes der Organisationsdatenbank. Insoweit sind die Daten eines Objektes und eines Datensatzes gleich. Der Unterschied besteht in der Art der Darstellung von Relationen: während die Datensätze der Datenbank reine Textbeschreibungen für diese Beziehungen haben, verwendet die Applikation schnellere Listen und Zeigerstrukturen. Die entsprechenden Konvertierungen geschehen nach dem Laden und vor dem Speichern einer Datenbank und werden in Kapitel 4.2.3 kurz erläutert.

Dabei existieren vier Listen, die jeweils Verweise auf alle Objekte eines Typs beinhalten. So gibt es Listen aller Personen, aller Abteilungen, aller Arbeitsgruppen und aller Rollen. Da in den Listen nur Verweise, aber keine Objekte stehen, kann die Applikation sie auf schnelle Weise sortieren. Dabei hat die Liste der Rollen eine Sortierung nach Namen und Parametern. Die Arbeitsgruppen sind nach dem Namen, die Personen nach Kombinationen aus Namen und Abteilungen sortiert. Die Abteilungen sind entsprechend ihrem Zuordnungspfad angeordnet.[154] Diese alphabetische Sortierung wird zur gesamten Laufzeit der Applikation aufrechterhalten. Wenn dann Listen der Reihe nach auf Objekte mit speziellen Attributen durchsucht werden, so sind die gefundenen Objekte alphabetisch sortiert.

Wie auch im Datenmodell stehen alle personenbezogenen Relationen generell bei den jeweiligen Personenobjekten.[155] Muß beispielsweise eine Arbeitsgruppe eine Liste ihrer Mitglieder erstellen, kann sie das nur über entsprechende Anfragen an alle

---

153 vgl. Kap. 9.6
154 vgl. Abbildung in Kap. 4.3.1.2.1
155 vgl. Kap. 3.2

Personenobjekte. Mit dieser Konvention hält auch die Applikation alle Daten streng redundant. Diese auf den ersten Blick umständliche Vorgehensweise bringt der Applikation die gleichen Vorteile wie der Datenbank: Einerseits ist es bei redundanter Datenhaltung leichter, die Korrektheit der Daten und Relationen sicherzustellen, andererseits ist der Speicherbedarf einzelner Objekte geringer und es können größere Datenbanken bearbeitet werden.
Der zur Laufzeit verfügbare dynamische Speicher ist besonders unter MS-Windows stark eingeschränkt,[156] so daß Datenredundanz wichtiger ist, als Geschwindigkeitsvorteile bei der Bearbeitung zu erreichen. Deshalb laufen einige Berechnungen langsamer ab, als es technisch möglich wäre. Spürbare Wartezeiten ergeben sich jedoch nur bei sehr großen Datenbanken, wenn innerhalb tabellarischer Darstellungen gescrollt wird.[157]

Neben den organisatorischen **Attributen** werden die elementaren Objekte mit graphischen oder organisatorischen Interaktionsmöglichkeiten, den sogenannten **Funktionen** belegt. Sie erhalten Informationen und Regeln dafür, wie sie sich bei Benutzeraktionen oder im Zusammenspiel verhalten sollen. Das kann z.B. im Verändern des Mauszeigers bestehen, solange dieser das entsprechende Objekt überstreicht. Allgemein ausgedrückt, werden Regeln festgelegt, nach denen sie ihren Zustand und ihre Relationen zu anderen Objekten nach einer Interaktion ändern sollen. So löst der Benutzer i.d.R. bei einem Objekt einen Vorgang aus, der daraufhin Auswirkungen auf eine Vielzahl anderer Objekte haben kann.

Beispiel:
Eine Interaktion besteht z.B. darin, daß eine graphisch dargestellte Arbeitsgruppe einen neuen Leiter bekommt. Der Vorgang beginnt in diesem Fall mit der objektorientierten Mitteilung an die Arbeitsgruppe, welches Personenobjekt der neue Leiter werden soll. Das Arbeitsgruppenobjekt sucht daraufhin nach einem bisherigen Manager, der erst auf Anfrage gelöscht werden kann. Der alte Manager ist jedoch dem Arbeitsgruppenobjekt nicht bekannt. Es muß vielmehr eine Anfrage an alle Personenobjekte senden, um ihn zu bestimmen. So werden u.U. alle Personenobjekte abgefragt und in den Interaktionsvorgang einbezogen. Gleichzeitig erhält das Objekt des potentiellen neuen Leiters eine Mitteilung, daß es nun die Arbeitsgruppe übernehmen soll. Damit werden in diesem Personenobjekt neue Aktionen gestartet usw..
Am Ende der gesamten Vorgänge ist möglicherweise ein Personenobjekt einer neuen Arbeitsgruppe zugeordnet und befördert, ein anderes ‘degradiert’ worden. Weiterhin kann sich die Größe zweier Arbeitsgruppenobjekte in der graphischen Darstellung ändern, da sie mehr oder weniger Mitarbeiter als vorher haben.

Dieses Beispiel beschreibt nicht alle Aktionen, die insgesamt stattfinden. Es zeigt hingegen, daß eine graphische, objektorientierte Darstellung allein schon in der Implementation der Datenstrukturen den realen Vorgängen relativ nahe kommt. Grundsätzlich besteht die interne Datenhaltung nicht aus tabellarischen Strukturen wie die Datensätze

[156] Bei der Verwendung von StarView Bibliotheken teilen sich ‘Stack’ und ‘Heap’ 64 Kilobyte. Der Stack wird für Funktionsaufrufe, der Heap für zur Laufzeit erzeugte Objekte benutzt.
[157] vgl. Kap. 4.5

der Datenbank. Vielmehr existieren relativ eigenständige Objekte, die nur durch graphische Interaktion oder mit Hilfe von Listen oder Zeigern angesprochen werden können. Der Vorgang des Ansprechens ist dabei recht wörtlich zu nehmen: Die Objekte können meist nicht passiv bearbeitet werden, sondern reagieren i.d.R. auf Nachrichten anderer Objekte. Die Vorteile einer solchen objektorientierten Datenhaltung bestehen in der recht einfachen Formulierung von Interaktionsregeln und der Möglichkeit zu realitätsnaher graphischer Bearbeitung durch den Benutzer. Sie verursacht jedoch eine nicht immer vorhersagbare Reihenfolge der ablaufenden Abgleiche und ist damit relativ schwer zu verifizieren.

Neben objektorientierten Interaktionsregeln bekommen Abteilungen und Arbeitsgruppen Informationen, wie sie sich in graphischer Darstellung zeichnen sollen.[158] Zusätzlich zu den elementaren Objekten werden bei Bedarf größere Objekte erzeugt, die die Darstellung globalerer Strukturen übernehmen. So ist beispielsweise ein Organigramm ein Objekt, das eine Liste aller Abteilungen, Attribute für Formatwerte, Farben usw. bekommt.

4.2.2 Funktionen zum Laden und Speichern

Zum **Laden** einer Datenbank hat der Benutzer drei Möglichkeiten:

1. Die letzten vier bearbeiteten Datenbanken sind im Menüpunkt ‘Datei’ eingetragen, und können durch einfache Auswahl geöffnet werden.
2. Beim Start kann eine Applikation eine Datenbank automatisch öffnen. Das ist entweder die zuletzt bearbeitete oder eine Standard - Datenbank, die in den Optionen der Applikation[159] angegeben wird.
3. Darüber hinaus besteht die Möglichkeit zur direkten Auswahl von Datenbanken über Verzeichnisstrukturen.

Bemerkung:
Bei jeder der Möglichkeiten können Datenbanken auch über Netzwerke geöffnet werden. Der Organization-Object Modeler ist jedoch nicht in der Lage, Verbindungen zu Servern aufzubauen. Deshalb kann er nur auf Server zugreifen, mit denen eine Verbindung besteht.

Das **Speichern** einer Datenbank erfolgt auf umgekehrtem Wege. Es existieren Menüpunkte zum direkten Speichern aller oder nur einer geöffneten Datenbank. Daneben erfolgt eine automatische Anfrage, wenn eine geänderte Datenbank geschlossen wird.

---

[158] sog. Paint - Routinen
[159] vgl. Kap. 4.4.5

Neben direktem Speichern gibt es eine komfortable Funktion für **automatische Speicherintervalle**. Dabei werden Zeitabstände bestimmt,[160] in denen eine Datenbank gesichert werden soll. Stellt der Organization-Object Modeler nach Ablauf eines Intervalls Änderungen fest, so speichert er auf Anfrage. Diese Anfrage erscheint direkt auf dem Bildschirm, auch wenn der Benutzer gerade mit einer anderen Applikation arbeitet. Es kann aber vorkommen, daß das Speicherintervall nicht exakt eingehalten wird, da die automatische Sicherung Leerlaufzeiten des Systems abwartet.

4.2.3 Konvertierungen beim Laden und Speichern

In der Notes - Datenbankumgebung werden Relationen zwischen organisatorischen Einheiten nur als Texte eingetragen. So steht im Datensatz einer Person der Name und hierarchische Zuordnungspfad ihrer Abteilung. Weitere Relationen, wie die Rollen und Arbeitsgruppen einer Person, werden lediglich als Textlisten angegeben.
Diese Art der Datenhaltung ist in Notes - Datenbanken zweckmäßig, da sie ein menschlicher Betrachter versteht und Datensätze relativ isoliert gehalten werden. Andererseits sind sie für eine Applikation zu langsam, da Textvergleiche viel Rechenzeit beanspruchen. Statt dessen verwendet der Organization-Object Modeler Zeigerstrukturen, die direkt auf ein spezielles Objekt verweisen.

Bei einem **Ladevorgang** generiert die Applikation zunächst für jede organisatorische Einheit ein Objekt. Jedes Objekt übernimmt die Relationen der jeweiligen Einheit zunächst in Textform. Erst nachdem die Applikation alle Objekte erzeugt hat, kann sie schnellere Zeiger darauf berechnen. Somit besteht der Ladevorgang einer Datenbank aus zwei Teilen: Zunächst aus dem vollständigen Einlesen aller Elemente, danach aus der Berechnung einzelner Zeiger.

Die Berechnung aller Zeiger an einem Stück ist relativ zeitintensiv. In der Entwicklungsphase der Applikation wurden zunächst möglichst alle Relationen durch Zeiger beschrieben. Dieser Lösungsweg mußte aufgegeben werden, da der Ladevorgang einer Datenbank inklusive Berechnung der Zeiger zu lange dauerte. Daraufhin wurde ein Kompromiß entwickelt, bei dem die Applikation nur eine begrenzte Anzahl von Zeigern benutzt, ohne daß große Zeitverluste entstehen.

Als Ergebnis transformiert der Organization-Object Modeler folgende Relationen in Zeiger:
- Die direkt vorgesetzte Abteilung einer Abteilung,
- eine Liste der Arbeitsgruppen, denen eine Arbeitsgruppe zugeordnet ist
- und die Abteilung einer Person.

Die Arbeitsgruppen zu denen eine Person gehört oder in denen sie Manager ist, werden nicht in Zeigern zugeordnet. Das erwies sich als notwendig, um einen schnellen

[160] vgl. Kap. 4.4.5

Bildschirmaufbau der Personentabellen zu erhalten, in denen die Arbeitsgruppen mit Namen angegeben werden.[161] Ein jeweiliges Sammeln der Arbeitsgruppen-Namen über einzelne Zeiger stellte sich als zu zeitaufwendig heraus. Auf doppelte Speicherung in Namenslisten und Zeigern wurde verzichtet, um vollständige Datenredundanz zu wahren und den Ladevorgang nicht weiter zu verlangsamen.

Da die Applikation Arbeitsgruppen und Rollen über ihre Namen identifiziert, müssen diese eindeutig sein. Abteilungen können gleiche Namen haben, aber nur bei unterschiedlichen Zeigern auf vorgesetzte Abteilungen. Diese Forderungen decken sich mit den Konventionen des Datenmodells.[162]

Beim **Speichern** ist eine Besonderheit der Notes - Datenbanken zu beachten: Einmal erzeugte Datensätze werden nicht physisch gelöscht, sondern existieren auch nach einem Löschbefehl vorerst weiter. Zwar existiert unter Notes eine Möglichkeit zum *Kompaktieren* einer Datenbank, die ‘gelöschte’ Datensätze physisch entfernt, das ist jedoch nicht in allen Umgebungen sinnvoll. Ein Vorhalten alter Datensätze soll Abgleiche zwischen verschiedenen Repliken einer Datenbank sicherer machen.
Zu beachten ist, daß eine Notes - Datenbank mit dem Erzeugen neuer Datensätze immer weiter anwachsen kann. Bei der Rückspeicherung in eine Datenbank sind die organisatorischen Objekte deshalb genau ihren Datensätzen zuzuordnen. Das geschieht auf recht einfache Weise, da jeder Datensatz[163] eine eindeutige Identifikationsnummer hat. Jedes Objekt der Applikation kennt die ‘Dokument - ID’ des entsprechenden Datensatzes und wird genau auf diesen Datensatz zurückgespeichert. Die IDs gelöschter Objekte merkt sich der Organization-Object Modeler in einer speziellen Liste, um danach die entsprechenden Datensätze ebenfalls zu löschen.

Die Speicherung geschieht objektweise in einem Durchgang, wobei die vorhandenen Zeiger wieder in Zeichenketten umgewandelt werden.

4.3 Repräsentation einer Datenbank mit Hilfe verschiedener Fenster

Wie in Kapitel 4.1.1 beschrieben, verwendet der Organization-Object Modeler verschiedene Datenfenster.[164] Es gibt unterschiedliche Typen von MDI - Fenstern für verschiedene Aspekte einer Aufbauorganisation. Sie werden im Rahmen dieses Kapitels systematisch erläutert.

[161] vgl. Kap. 4.3.1.2.3
[162] vgl. Kap. 3.2
[163] in der Notes - Terminologie ein ‘Dokument’
[164] MDI - Fenster

Es können MDI - Fenster verschiedenen Aufbauorganisationen gleichzeitig geöffnet sein. Damit diese potentielle Vielzahl von Fenstern nicht unübersichtlich wird, wurde für den Benutzer ein spezieller Aufbau entwickelt:

Da die meisten Menschen bei einer Unternehmensorganisation an ein Organigramm denken, präsentiert sich eine Organisationsdatenbank mit einem Hauptfenster, in der die Abteilungshierarchie graphisch dargestellt ist.[165] Diesen Fenstertyp beschreibt Kap. 4.3.1.1 . Wird das Hauptfenster geschlossen, schließt sich die gesamte Datenbank mit allen evtl. erzeugten, anderen MDI - Fenstern. Vom Hauptfenster aus können weitere Fenster geöffnet werden. Es gibt eines zur graphischen Darstellung von Abteilungsbeziehungen und vier Tabellendarstellungen für Personen, Abteilungen, Arbeitsgruppen oder Rollen. Sie werden in Kapitel 4.3.1.2 bzw. Kap. 4.3.2 vorgestellt.

Jedes der MDI - Fenster kann als Sinnbild klein geschaltet werden.[166] Neben diesem vollständigen Wegblenden eines Fensters existieren für graphische Darstellungen verschiedene Formatierfunktionen, unter anderem zum maßstabsgerechten Verkleinern. Das hat den Vorteil, daß der Benutzer bei kleineren Fenstern noch einen Überblick über organisatorische Strukturen sehen kann. Sämtliche Formatierfunktionen erläutert Kapitel 4.3.1.3 .

4.3.1 Graphische Darstellungen

4.3.1.1 Das Organigramm - Abteilungen in graphischer Interaktion

Ein MDI - Fenster dient zur Darstellung und Interaktion mit einer Abteilungshierarchie. Es präsentiert die Abteilungen als graphische Objekte, die mit eigenen Funktionalitäten ausgestattet sind. Die graphische Darstellung entspricht den Vorgaben aus Kapitel 3.3.2.2.1.
Die Hierarchie der Aufbauorganisation wird ‘Top-Down’ oder ‘Bottom-Up’ aufgebaut. Dabei stellt die oberste Aggregationsebene das Unternehmen selbst dar. Darunter folgen ebenenweise die Unterabteilungen in beliebiger Aggregation:

[165] z.B. das linke obere MDI - Fenster der Abbildung 4-1
[166] sogenanntes ‘Ikonisieren’

Abbildung 4-2 Abteilungshierarchie in einem MDI - Fenster

Die kleinen weißen Felder reagieren auf Mausklick und verbergen oder expandieren Unterstrukturen. Ein Minuszeichen steht für eine sichtbare, ein Pluszeichen für eine verborgene Unterstruktur. Beim Überstreichen mit dem Mauszeiger zeigt ein Aggregati onsfeld die Zahl der Unterabteilungen an.

Ungebundene Abteilungen werden in einem separaten Fenster, dem *Abteilungs - Pool* gesammelt.[167] Das sind Abteilungen, die in keiner hierarchischen Beziehung zu anderen Abteilungen stehen. Dieses Sammeln ist notwendig, da sie ansonsten auf oberster Ebene der graphischen Darstellung stünden, wo sie schwieriger zu lokalisieren wären.

Der Benutzer kann Abteilungen in der Abteilungshierarchie beliebig in Beziehung setzen. Dazu werden sie mit der Maus entweder auf eine freie Stelle oder eine andere Abteilung gezogen. Die möglichen Bewegungen von Abteilungsobjekten haben unterschiedliche Auswirkungen:

- Eine Abteilung wird durch direktes Ziehen Unterabteilung einer anderen.
- Der Benutzer kann eine Abteilung unterhalb ihrer Oberabteilung beliebig verschieben und verändert damit ihre hierarchische Ebene.[168]
- Verschiebt der Benutzer eine Abteilung hingegen höher, wird sie selbständig und auf oberster Ebene angeordnet.
- Einzelne ungebundene Abteilungen aus dem Abteilungs - Pool können auf oberster Ebene angeordnet werden.

[167] in der Abbildung rechts zu sehen
[168] auffallend bei der Abteilung ‘Posteingang’ in Abbildung 4-2

- Zieht der Benutzer eine Abteilung auf den Abteilungs - Pool, verliert diese alle Relationen und ihre bisherigen Unterabteilungen steigen eine Hierarchiestufe empor oder kommen ebenfalls in den Abteilungs - Pool.
- Neue Abteilungen werden erzeugt, indem der Benutzer das oberste Symbol des Abteilungs - Pools in das Organigramm zieht.

Über die Beziehungen zwischen Abteilungsobjekten und dem Organigramm hinaus sind Verschiebungen von Personen auf Abteilungen möglich. Eine Verschiebung in die oberste Hälfte ernennt eine Person zum Abteilungsleiter, ansonsten werden der oder die Personen einfache Mitglieder. Das ist in folgender Abbildung zu sehen: die linken Personen werden Mitglieder und die rechte Person Abteilungsleiter der Abteilung 'Entwicklung':

Abbildung 4-3 Ziehen von Personen auf eine Abteilung

Für Abteilungshierarchien bietet der Organization-Object Modeler eine Vielzahl von Formatiermöglichkeiten, die das Aussehen anpassen oder verändern können.[169]

4.3.1.2 Arbeitsgruppen in Beziehung gesetzt - die Arbeitsgruppen 'Netzwerke'

Ein weiterer Fenstertyp stellt Beziehungen von Arbeitsgruppen dar. Er verwendet eine angedeutete dreidimensionale Perspektive, die in Kapitel 3.3.2.2.2 entwickelt wurde:

[169] vgl. Kap. 4.3.1.3

Abbildung 4-4 Arbeitsgruppendarstellung in einem MDI - Fenster

Die gestrichelten Linien sollen informelle Beziehungen andeuten, die zwischen den Arbeitsgruppen bestehen. Sie besagen, daß eine Arbeitsgruppe über die Arbeitsgruppen verfügen kann, die in der Darstellung unter (bzw. vor) ihr liegen. So ist z.B. die Arbeits-gruppe ‘not named7’ eine Teilgruppe sowohl von ‘Produkttest’ als auch von ‘not named4’.

Auch bei den Arbeitsgruppen ergeben sich verschiedene Interaktionsmöglichkeiten:

- Zieht der Benutzer eine oder mehrere Arbeitsgruppen auf eine Arbeitsgruppe, so integriert er sie darin.
- Bewegt der Benutzer den Mauszeiger in die Nähe einer Verbindungslinie, wandelt er sich in eine Schere. Auf Klick wird damit die Relation aufgelöst.
- Arbeitsgruppen aus anderen Organisationen kann der Benutzer in das Fenster kopieren oder verschieben. Auf Rückfrage übernimmt die Applikation dabei auch die Gruppen-mitglieder.
- In Arbeitsgruppen sind die Mitglieder graphisch dargestellt. Sie können beliebig zwischen Arbeitsgruppen verschoben oder kopiert werden. Ein Kopiervorgang integriert Personen in eine neue Arbeitsgruppe, entfernt sie aber nicht aus der alten. Auf diese Weise können auch Personen zwischen verschiedenen Datenbanken ausge-tauscht werden.

Der Organization-Object Modeler erlaubt nur Verknüpfungen von bis zu 5 Arbeitsgrup-pen in einer direkten Linie.[170]

[170] vgl. Kap. 3.1.2

Der Benutzer kann eine Person an verschiedene Stellen einer Arbeitsgruppe ziehen. Bei Zug auf die obere Hälfte macht er sie zum offiziellen Gruppenleiter, in der unteren Hälfte zum normalen Mitglied.

Anmerkung:
Die Darstellung der Arbeitsgruppenbeziehungen hat keine streng dreidimensionale Perspektive:
- Zum einen müßten die Personen an der Vorderseite einer Arbeitsgruppe weiter auseinanderstehen als hinten. Die vorderen Personen konnten mit Hilfe mehrerer Ellipsengleichungen aber nur geringfügig auseinandergezogen werden.
- Desweiteren bauen dreidimensionale *Perspektiven* auf einer Fluchtpunktdarstellung auf. Dabei verjüngt sich die Darstellung auf einen Fluchtpunkt im Hintergrund. Das hat jedoch zur Folge, daß Arbeitsgruppen, die weiter hinten liegen, kleiner gezeichnet werden müßten. Eine Berechnung der dreidimensionalen Lage jedes Objektes würde bei Benutzerinteraktionen zu lange dauern. Deshalb wurde anstelle einer **Perspektivprojektion** eine **Parallelprojektion** verwendet, die weiter entfernt liegende Teile unnatürlich groß wirken läßt.[171]

4.3.1.3 Formatierfunktionen und deren Verwendungsmöglichkeiten

In den vorhergehende Kapiteln wurden die MDI - Fenster für Organigramme und Arbeitsgruppenbeziehungen vorgestellt. Mit Hilfe der Property - Box[172] können diese Darstellungen auf verschiedene Arten verändert werden. Zunächst folgt eine Beschreibung der Formatierung von Abteilungs-, anschließend eine der Arbeitsgruppengraphen. Der Benutzer kann jedes MDI - Fenster unterschiedlich formatieren.

Organigramme

Formatierungen werden mit folgenden ‘Seiten’ der Property - Box vorgenommen. Das ist zum einen die Seite für Formatierung[173], zum anderen eine Seite zum Einstellen der Hintergrundfarbe.

[171] vgl. z.B. **[Hartwig1987]** S. 199
[172] vgl. Kap. 4.4.5
[173] vgl. Abbildung 4-5

Abbildung 4-5: Formatierung eines Organigramms

Die Formatierseite hat links ein Vorschaufenster, in dem Änderungen direkt zu sehen sind. Darunter kann der Benutzer einstellen, ob Änderungen direkt auch in das MDI - Fenster übertragen werden, oder ob die Einstellungen von nun an für alle Organigramme gelten sollen.

Die eigentlichen Funktionen stehen auf der rechten Seite:

- **Ausrichtung** Ordnet die Abteilungen in einer vertikalen oder horizontalen Pyramide an.
- **Rastergröße** Bestimmt die horizontalen und vertikalen Abstände zwischen Abteilungen in Bildpunkten.
- **Zoom:** Vergrößert oder verkleinert die Darstellung. Das ist besonders zur Navigation in großen Strukturen interessant, weil alle Funktionalitäten erhalten bleiben.
- **Ebenen:** Hier kann die Anzahl der expandierten Ebenen begrenzt werden. Das ist besonders zum Einstieg in große Organigramme sinnvoll.

Als Hintergrundfarbe sind alle verfügbaren Farbnuancen oder Farbmischungen zugelassen.

Arbeitsgruppen

Die Darstellung der Arbeitsgruppen kann der Benutzer mit einer Teilmenge der gleichen Funktionen formatieren. Sie sind in folgender Abbildung zu sehen:

Abbildung 4-6 Formatierung einer Arbeitsgruppendarstellung

Wie bei der Formatierung von Organigrammen gibt es ein Vorschaufenster, das direkt auf Änderungen reagiert. Da die Darstellung von Arbeitsgruppen aufwendiger zu berechnen ist, kann das Vorschaufenster etwas langsam werden. Der Benutzer kann es deshalb bei Bedarf ausblenden. Die Animation der Arbeitsgruppen kann mit dem mittleren Kontrollkästchen ausgeschaltet werden. Dann haben alle Arbeitsgruppen dieselbe Größe und keine Personensymbole.

Anmerkung:
Die Animation *einzelner* Arbeitsgruppen wird direkt in ihrem Kontextmenü an- und ausgeschaltet. Das ist von Vorteil, wenn z.B. einzelne große Gruppen zuviel Platz beanspruchen.

Anmerkung:
Grundsätzlich sind **beide** Graphikformen beliebig verkleiner- und vergrößerbar. Dazu benutzt der Organization-Object Modeler Vektorgraphiken und skalierbare Schriftarten. Er kann aber nur Schriftarten verwenden, die das Betriebssystem zur Verfügung stellt. Fehlt eine Schriftart oder eine Größe einer Schriftart, wird sie substituiert. Deshalb kann es vorkommen, daß beim Zoomen Beschriftungen eine etwas unpassende Größe bekommen.

4.3.2 Der Browser - Daten in tabellarischer Form

Neben der graphischen gibt es die Möglichkeit einer tabellarischen Darstellung und Bearbeitung der Datenbestände. Dazu wurde ein Werkzeug entwickelt, das verschiedene Interaktionen auch mit den graphisch dargestellten Objekten ermöglicht. Der sog. **'Browser'** ist eine Neuimplementierung und Weiterentwicklung einer StarView - Bibliothek,[174] die sich als zu langsam und unzuverlässig erwiesen hat. Er dient zur zeilenweisen Darstellung von Personen, Abteilungen, Arbeitsgruppen oder Rollen. Die tabellarische Darstellung bietet Möglichkeiten zum schnellen Bearbeiten von Einträgen oder Eingeben neuer Objekte.
Im Organization-Object Modeler ist für Personen und Rollen keine direkte graphische Darstellung vorgesehen, so daß ein leistungsfähiges Werkzeug zu ihrer Bearbeitung erschaffen werden mußte. Die Elemente in den verschiedenen Browser - Tabellen können auch unter Lotus Notes direkt bearbeitet werden, dort steht jedoch eine dokumentenweise Bearbeitung im Vordergrund, nicht aber eine direkte Manipulation in Tabellen.

Zunächst wurde ein generischer Browser erschaffen, der sich für die beliebige Darstellung tabellarischer Daten eignet. Der Organization-Object Modeler benutzt diesen universellen Browser mit einer Teilmenge der Funktionalitäten in vier Abwandlungen für die vier Grundelemente der Organisationsdatenbank.

Der allgemeine Aufbau des Browsers besteht aus Zeilen, die jeweils ein Objekt darstellen. Die Attribute der Objekte stehen in den Spalten, die in ihrer Breite vom Benutzer variierbar sind. In einem festen Bereich des Datenfensters werden auf der linken Seite Spalten gezeigt, die zur Identifikation der Objekte wichtig sind.[175] Im rechten Teil des Datenfensters sind Spalten frei scrollbar.
Eine besondere Funktionalität bieten die Zeilenknöpfe. Man kann sie an- und ausstellen, um Objekte zu selektieren. Hierbei gibt es auch die Möglichkeit von Mehrfachselektionen.

---

[174] vgl. **[StarView1992c]**
[175] beim Personen-Browser z.B. der Name

Abbildung 4-7: Der Browser

Selektierte Zeilen können mit Menübefehlen oder Knöpfen der Werkzeugleiste gelöscht oder kopiert werden. Desweiteren dienen die Zeilenknöpfen als Ansatzpunkte für Drag&Drop - Vorgänge, um Objekte des Browsers in andere Browser oder graphische Darstellungen zu kopieren oder zu verschieben. Darüber hinaus kann der Browser das Ziel von Drag&Drop - Vorgängen sein. Unterschieden wird, in welcher Zeile und Spalte der Drop - Vorgang stattgefunden hat.

Alle Felder der verschiedenen Browser können auch über Auswahlboxen oder variable Textfenster bearbeitet werden. Diese Fenster schalten sich ein, sobald ein Texteintrag bei der Eingabe länger als das Feld wird. So ist es z.B. nicht nötig, alle Mitglieder einer Abteilung über Tastatur einzugeben. Ihre Namen werden einfach in einer Listendarstellung an- und ausgeschaltet.

Der konkrete Aufbau und die einzelnen Funktionen[176] der Browser werden in den nächsten Unterkapiteln beschrieben.

Bemerkung:
Ein Detail ist für den technisch versierten Leser interessant: Der Browser stellt keine Speicherbereiche für die einzelnen Felder der Tabelle bereit, sondern liest oder berechnet ihre Inhalte aus den Datensätzen. Das geschieht jedes Mal, wenn in der graphischen Benutzeroberfläche ein Teilbereich eines Browser - Fensters neu gezeichnet werden muß, beispielsweise beim Scrollen.
Dieses Merkmal stellt einen Kompromiß zwischen Schnelligkeit des Bildaufbaus und sparsamer Speicherausnutzung dar. Besonders unter MS-Windows ist der zur Laufzeit dynamisch verfügbare Speicher begrenzt[177], so daß sparsame Speicherauslastung vorgezogen wurde, um größere Datenbanken bearbeiten zu können.

[176] auch einige der Drag&Drop Beziehungen
[177] hauptsächlich 64 Kilobyte bei der Verwendung von StarView Bibliotheken

4.3.2.1 Abteilungen

Der Abteilungs - Browser enthält Abteilungen und ihre Attribute. Neben den in Kap. 3.3.2.1.2 beschriebenen Attributen werden für jede Arbeitsgruppe die Mitglieder, die Größe und der Manager dargestellt.

Abteilungen von:"Sportartikel und Heimtierbedarf"

Unterabteilung vo	Name	Manager	Stabstelle	#	Mitglieder	Beschreibung
	ACME		nein	0		
ACME\	Development		nein	2	Charly Brown, Jo	
ACME\	Global Assekura		nein	0		
ACME\Global Ass	Niederlassung O		nein	0		
ACME\Global Ass	Posteingang		nein	5	Heiner Müller, Kl	Konzeption und E
ACME\	Production		ja	0		
ACME\	Sales		nein	3	Beavis, Bernhard	
ACME\Sales\	Europe		nein	1	Bob Marley	
ACME\Sales\	North America		nein	1	Ray Parker	
	Pavone		nein	3	Anne Ban, Charle	
Pavone\	test		nein	2	Amie Stewart, Be	
	Wasser - Rugby		nein	0		
Wasser - Rugby\	Schwimmen		nein	0		Wettkämpfe orga

Abbildung 4-8 Der Abteilungs - Browser[178]

Die direkte Eingabe über Tastatur wird im Abteilungs - Browser durch einige Zusatzfenster unterstützt. Das sind Textfenster für das Editieren einer Beschreibung, eine Listendarstellung für schnelle Auswahl der Mitglieder oder eines Managers und erweiterte Textfelder für einzeilige Eingaben.

Der Abteilungs - Browser ist mit vielfältigen Drag&Drop - Beziehungen ausgestattet. Diese definieren, an welcher Stelle Objekte ‘fallengelassen’ werden können und was das für Auswirkungen hat. Einzelne Abteilungen des Abteilungs - Browsers werden über die Zeilenknöpfe ‘angefaßt’ und verschoben oder kopiert.[179] Verschiebe- und Kopiervorgänge unterschieden sich dadurch, ob eine systemspezifische Taste gedrückt gehalten wird.[180] Personen werden bei einem Drag&Drop - Vorgang entweder Abteilungsleiter, wenn man sie über der Spalte ‘Abteilungsleiter’ fallen läßt, ansonsten Mitglieder. Falls sie in der Datenbank des Abteilungs - Browsers noch nicht definiert sind, generiert die Applikation sie nach Rückfrage. Zu beachten ist, daß Personen nur zu einer Abteilung gehören können. Sind sie bereits in einer anderen Abteilung, kommt eine entsprechende Rückfrage.

[178] schmalere Darstellung als original
[179] Werden Abteilungen zwischen verschiedenen Datenbanken verschoben, erscheint eine Anfrage, ob sie mit oder ohne Mitarbeitern eingefügt werden sollen. Unbekannte Personen generiert die Applikation auf Einzelanfrage neu.
[180] bei Windows95 die <Shift> Taste

Unbekannte Abteilungen werden auf Anfrage hinzugefügt, bekannte Abteilungen der entsprechenden Abteilung untergeordnet.[181]

4.3.2.2 Arbeitsgruppen

Die tabellarische Darstellung der Arbeitsgruppen ist nahezu identisch mit der Darstellung von Abteilungen. Es gelten die gleichen Drag&Drop - Funktionen, wie im vorhergehenden Kapitel beschrieben.[182] Da es keine eigentliche Hierarchie gibt, sind die Arbeitsgruppen nach ihrem Namen sortiert.

Arbeitsgruppen von: "Sportartikel und Heimtierbedarf"

Name	ist integriert in	Manager	#	Mitglieder	Beschreibung
Nachrichten	Wetter	Beavis	3	Beavis, Charles (	sdfsdf
not named			0		
not named4	not named		0		
not named5	Produkttest	John Brown	2	John Brown, Nad	
not named6	not named4		0		
not named7	not named4		1	Ray Parker	
not named8	not named6		0		
Produkttest	Verkehrsentwicklung Süd	Bob Marley	17	Amie Stewart, An	
Verkehrsentwickl		Heiner Müller	4	Heiner Müller, Kl	
Wetter			1	Berti Vogts	

Abbildung 4-9: Der Arbeitsgruppen - Browser

4.3.2.3 Personen

Die aufwendigste Tabellendarstellung ist der Personen - Browser. In ihr stehen alle Personen einer Datenbank mit ihren Attributen. Feste Spalten bilden Name und Vorname der Person, während die anderen Spalten gescrollt werden können.
Im Personen - Browser sind die Personen wahlweise nach Namen oder Abteilung[183] alphabetisch sortiert.

---

[181] **Beispiel:** Würde die Abteilung 'Schwimmen' in der letzten Zeile aus Abbildung 4-8 angefaßt und auf der ersten Zeile fallengelassen, würde 'Schwimmen' neue Unterabteilung von 'ACME', nicht mehr von 'Wasser-Rugby'.
[182] vgl. Kap. 4.4.2.2.2
[183] Abbildung 4-10

Personen von: "Sportartikel und Heimtierbedarf"

Abteilung	Nachname	Vorname	Initial	Certifier	Abteilungsleiter	Arbeitsgruppen	Arbeitsgruppenleiter	Rollen
	Huth	Carsten			nein	Produkttest		
ACME\Developr	Brown	Charly			nein			
ACME\Developr	Brown	John			nein	not named5	not named5	Hotline/E
ACME\Global A:	Krause	Klaus			nein	Produkttest, Verk		Hotline/N
ACME\Global A:	Müller	Heiner			nein	Verkehrsentwickl	Verkehrsentwicklung	Hotline/N
ACME\Global A:	Wussow	Klaus Jürgen			nein	Produkttest		

Abbildung 4-10 Der Personen - Browser[184]

Personen können vielfältiges Ziel einer Drag&Drop - Aktion sein. Ihnen werden
- Abteilungen,
- Arbeitsgruppen
- oder Rollen zugewiesen.

Wird eine Abteilung auf der Abteilungsleiter-Spalte ‘fallengelassen’, so wird diese Person Leiter der Abteilung, ansonsten Mitglied. Bei Arbeitsgruppen geschieht das gleiche, nur daß mehrere Arbeitsgruppen gleichzeitig zugewiesen werden können. Rollen werden einfach über der entsprechenden Zeile einer Person ‘fallengelassen’.

4.3.2.4 Rollen

Die einfachste Tabellendarstellung ist der Rollen - Browser. In ihr stehen alle Rollen mit ihren Attributen in alphabetischer Reihenfolge.
Selektierte Rollen werden kopiert, verschoben oder gelöscht. Durch Verschieben oder Kopieren können Rollen in andere Datenbanken übertragen werden. Ein anderes Ziel können ein oder mehrere Personen sein, die damit die Rollen erwerben.

Rollen von: "Sportartikel und Heimtierbedarf"

Name	Parameter	Inhaber	#	Beschreibung
Cannes Rolle	silberne Palme		0	
Hotline		Amie Stewart, Anne Ban, Be	4	
Hotline	Europe	John Brown, Ray Parker	2	Telefonische Produktberatung ...
Hotline	North America	Heiner Müller, John Brown, I	3	
Programmierer			0	
Programmierer	Pascal		0	

Abbildung 4-11 Der Rollen - Browser

Der Rollen - Browser dient nur bei wenigen Drag&Drop - Aktionen als Ziel. Rollen werden aufgenommen, falls sie noch nicht existieren und Personen werden Inhaber der entsprechenden Rolle.

[184] schmalere Spalten als original

4.4 Beschreibung besonderer Konzepte der Applikation

4.4.1 Anlegen von Bibliotheken und deren Verwendung

Es wäre relativ unproduktiv, mit jeder neuen Datenbank alle Strukturen von Grund auf neu zu entwickeln oder einzugeben. Gewöhnlich sind Organisationsstrukturen bereits in anderen Datenbanken vorhanden und im praktischen Einsatz getestet worden. Mit alten Datenbanken sammeln sich deshalb im Laufe der Zeit verschiedene Aufbaustrukturen an. Sie repräsentieren einen Teil des Wissens, den eine Unternehmung im Umgang mit ihrer Aufbauorganisation gesammelt hat. Eine neue Aufbauorganisation wird meist ein Stück besser sein, als die vorherige. Somit stellen alte Organisationsformen einen reichhaltigen Erfahrungswert dar, den es zu nutzen gilt.

Bei dynamischer Veränderung einer Aufbauorganisation werden sich verschiedene Teilstrukturen als besonders effizient erweisen. Andere Teilstrukturen sind aufgrund vielfältiger Überlegungen oder Erfahrungswerte entstanden. Für eine rationelle Entwicklungstätigkeit könnten diese Strukturen identifiziert und als Module abgelegt werden. Bei größeren Unternehmen bietet sich der Aufbau einer Bibliothek mit verschiedenen thematischen Datenbanken an, deren Module generisch in neue Organisationsstrukturen eingefügt werden können.

Der Organization-Object Modeler unterstützt solche modulare Wiederverwendbarkeit, indem er mehrere Möglichkeiten zum Austausch beliebiger Strukturen zwischen verschiedenen Datenbanken bereitstellt. Dafür werden Datenbanken geöffnet und Strukturen mit oder ohne die betroffenen Personen übertragen. Das geschieht entweder mit der **Zwischenablage** oder durch **Drag&Drop - Aktionen** Die Zwischenablage dient zum Kopieren von Abteilungshierarchien, verknüpften Arbeitsgruppen, Personen oder Rollen. Daneben kann der Benutzer markierte Einheiten direkt mit der Maus in andere Datenbanken übertragen. Sie werden dabei kopiert oder verschoben, abhängig davon, ob eine dafür übliche Taste gedrückt wird. Bei Drag&Drop - Aktionen werden allerdings alle organisatorischen Objekte gleich behandelt, d.h. es werden keine Abhängigkeiten oder Hierarchien übertragen.
Eine formale Beschränkung ergibt sich bei Namensgleichheiten, da dann neue Einheiten umbenannt werden müssen. Die Applikation weist jedoch in diesen Fällen darauf hin und zeigt eine Eingabezeile zur direkten Namensänderung.

Ein Ganzes ist mehr als seine Teile: Insgesamt besteht die Gefahr, daß eine Zusammensetzung einzelner, an sich effizienter, Teilmodule eine fehlerhafte Gesamtstruktur hervorbringt. Diese Gefahr wird aber relativiert, da die Mitglieder des organisatorischen Systems eigenverantwortlich mit einer dynamischen Veränderung reagieren können. Da aber Mitarbeiter meist nur lokale Gegebenheiten überschauen können, sollte jedoch dieser Aspekt bei der Verwendung von Bibliotheken nicht außer acht gelassen werden.[185]

---

[185] In gewissem Sinne verwendet der Organization-Object Modeler auch Notes - Adreßbücher als Bibliotheken (in der Notes - Terminologie ‘Name & Adressbooks’). In ihnen definierte Personen oder

Neue Organisationsdatenbanken können bisher nicht angelegt werden. Entsprechende Funktionen sind implementiert, aber in den Quelldateien auskommentiert und dadurch nicht funktionsfähig. Diese Beschränkung ist durch zwei Überlegungen entstanden:
Zum einen kann der Organization-Object Modeler zwar leere Datenbanken erzeugen, diese besitzen jedoch alle dieselbe Replikations - ID. Da Lotus Notes anhand dieser IDs Datenbanken abgleicht, also unbekannte Datensätze austauscht, wären die neu erzeugten Datenbanken bisher nicht voll verwendungsfähig. Würde eine Möglichkeit zum Setzen der Replikations - ID geschaffen, könnten auch neue Datenbanken erzeugt und verwendet werden. Diese Funktion wird zur Zeit von der Macroware - DLL[186] nicht unterstützt.
Ein zweiter Grund ist eine Überlegung der Firma Pavone, die aus strategischen Überlegungen das Anlegen neuer Datenbanken nicht ermöglichen möchte.

4.4.2 Kontrollfunktionen zur Wahrung der Konsistenz

Der Organization-Object Modeler bietet verschiedene Funktionen zur Wahrung der Konsistenz einer Datenbank. Was eine konsistente Datenbank sein soll, wird weiter unten anhand konkreter Testaktionen erläutert. Die Applikation unterscheidet zwischen zwei Arten von Inkonsistenzen: den offensichtlichen **Fehlern**, die eine Konsistenz der Datenbank direkt zerstören, oder **geringfügige Inkonsistenzen**, die meist den Regeln des Datenmodells widersprechen, jedoch eine Weiterarbeit erlauben. Die Abstufung in Fehler und Warnungsmeldungen hat ihre Berechtigung, da beim Erstellen einer Organisationsdatenbank nicht sofort alle Elemente und Zuordnungen dem Organisationsmodell entsprechen. So kann es z.B. vorübergehend leere Abteilungen oder Arbeitsgruppen geben.

Jede Interaktion des Benutzers wird direkt auf Wahrung der Konsistenz untersucht. Liegt eine unerlaubte Handlung vor, zeigt die Applikation eine Fehlermeldung und macht die letzte Aktion rückgängig bzw. erlaubt sie erst gar nicht. Neben der direkten Überwachung von Benutzeraktivitäten existiert ein Modul zur Fehlersuche, das der Benutzer direkt aufrufen kann. Es erzeugt einen Report mit Fehlern und Warnungen.

Eine explizite Suche nach Fehlern ist besonders nach dem Laden einer Datenbank sinnvoll, da in der Zwischenzeit unter Notes Datensätze gelöscht sein können oder durch Datenbankabgleiche u.U. doppelt vorkommen. Warnungsmeldungen hingegen beschreiben geringfügige Inkonsistenzen. Vor dem Praxiseinsatz einer Datenbank sollten auch diese Mängel beseitigt werden. Somit ist eine Suche nach den geringfügigeren Inkonsistenzen beim Speichern einer fertigen Datenbank sinnvoll.
Der Organization-Object Modeler kann deshalb automatische Tests direkt nach dem Laden oder vor dem Speichern einer Datenbank ausführen. Standardmäßig sind diese

---

Arbeitsgruppen listet er in sog. **Drag&Drop - Werkzeugen** auf. Aus diesen Listen kann der Benutzer die Adreßbuch-Elemente durch einfaches Ziehen mit der Maus in eine Datenbank einbringen (vgl. Kap. 4.4.4).

[186] vgl. Kap. 4.1.2

Tests angeschaltet. Werden Fehler gefunden, zeigt er einen ausführlichen Fehlerbericht und eine Statistik der Datenbank.

Der Benutzer kann verschiedene Tests ablaufen lassen, die nur Personen, Abteilungen, Arbeitsgruppen oder Rollen betreffen. Das verringert bei großen Datenbanken die Laufzeit der Kontrolle. Ein mögliches Ergebnis einer Konsistenzprüfung könnte z.B. wie folgt aussehen:

Abbildung 4-12 Meldungen einer Konsistenzprüfung

Auf einer Organisationsdatenbank werden folgende Tests angewendet:

a) folgende 'Tatbestände' verursachen **Fehlermeldungen**:
- Kreise in der Abteilungshierarchie[187],
- Verknüpfung von Arbeitsgruppen über mehr als fünf Stufen in einer Richtung,
- doppelte Namensvergabe bei Personen und Arbeitsgruppen,
- Rollen mit gleichem Namen und Parameter,
- Abteilungen, die gleichen Namen haben und derselben Abteilung unterstellt sind,
- Abteilungen, Arbeitsgruppen oder Rollen, die referenziert sind, aber nicht existieren.

b) Diese Inkonsistenzen werden als **Warnungen** erwähnt:
- Abteilungen ohne Abteilungsleiter,
- leere Abteilungen oder Arbeitsgruppen,

[187] eine Abteilung kann nicht Unterabteilung von sich selbst sein

- Personen, die keiner Abteilung angehören,
- Rollen, die keiner Person zugeordnet sind.

Eine genauere Beschreibung von Fehlern oder Warnungsmeldungen zeigt die Applikation im Rahmen einer größeren Statistik.[188] Die Statistik wird im folgenden Kapitel erläutert.

4.4.3 Statistiken erstellen

Statistiken ermöglichen dem Benutzer verschiedene Sichtweisen auf eine Aufbauorganisation. Sie unterstützen verschiedene aufzählende Fragestellungen. So gibt es neben der reinen Beschreibung von Fehlern noch eine Reihe anderer Funktionen, die Elemente mit bestimmten Merkmalen auflisten können.

Als erstes folgt eine Beschreibung der Statistik für **Abteilungen**. Die Statistik wird entweder direkt durch einen Knopf der Toolbox oder einen Menübefehl erstellt oder zur detaillierteren Fehlerbeschreibung nach einem Konsistenztest angezeigt.

Datenbank - Statistik
Abteilungen ... Arbeitsgruppen Personen ... Rollen ...
Fehler:
total = 12 in 4 Ebenen: 2 Führungsstellen
9 ausführende Stelle(n)
1 Stabstelle(n)
Abteilungen mit 0 Mitgliedern (leer): 6
Abteilungen ohne Manager: 12
ACME
ACME\Einkauf
ACME\Einkauf\Niederlassung Ost
ACME\Entwicklung\unbenannt
ACME\Entwicklung\unbenannt\unbenannt2
ACME\Rechtsfragen

Abbildung 4-13 Statistik der Abteilungen einer Aufbauorganisation

Für **Arbeitsgruppen** existiert eine nahezu identische Statistik, die aber keine Arbeitsgruppen ohne Manager behandelt.

---

[188] Knopf ‘Details’ in der Abbildung 4-12

In der Statistik für **Personen** sind alle Personen ohne Abteilung, alle Abteilungsleiter oder alle Arbeitsgruppenleiter zu sehen.

Bei der Statistik von **Rollen** kann der Benutzer Rollen anzeigen lassen, die x - viele Inhaber haben. So ist z.B. die Fragestellung interessant, welche Rollen genau einer Person zugeordnet sind.

Wenn doppelte Einheiten vorkommen, wie z.B. Arbeitsgruppen mit gleichem Namen, dann wird bei allen Statistikfenstern ein weiteres Feld zur Auflistung dieser Einheiten angezeigt.

Beispiel:
In Abbildung 4-13 ist der Platz für doppelte Abteilungen oberhalb der Liste grau, da es keine doppelten Abteilungen gab.

Anmerkung:
Auf *eine* Person bezogene Informationen können in der kontextsensitiven Property - Box[189] angezeigt werden. Sie geben u.a. Listen mit Arbeitsbeziehungen zu anderen Personen an. Dabei handelt es sich um
- direkte und indirekte Vorgesetzte,
- direkte und indirekte Mitarbeiter,
- und direkte und indirekte Untergebene.[190]

In Abbildung 4-14 sind z.B. links die Personen zu sehen, mit denen eine Person in ihrer Abteilung zusammenarbeitet. In der rechten Liste stehen alle Personen, die mit der Person direkt oder indirekt über verknüpfte Arbeitsgruppen zusammenarbeiten.

[189] vgl. Kap. 4.5.5
[190] nur bei Managern

Abbildung 4-14 Arbeitsbeziehungen einer Person

4.4.4 Schnelle Zuweisung über ein oder mehrere Drag&Drop Werkzeuge

Die sogenannten ‘Drag&Drop - Werkzeuge’ ermöglichen eine schnelle Vorgehensweise zur Zuweisung von Objekten. Sie zeigen kontextabhängig verfügbare Objekte, die aus Listen heraus auf andere Objekte gezogen werden können. In Abbildung 4-15 werden auf diese Weise Personen auf Abteilungen verteilt:

Abbildung 4-15 Schnelle Zuweisung über mehrere Drag&Drop Werkzeuge

Das *linke* Drag&Drop Fenster zeigt Personen an, die derzeit keiner Abteilung angehören, also *ungebunden* sind. Wird eine Person auf eine Abteilung gezogen, verschwindet sie aus der Liste. Das erweist sich als besonders hilfreich zum Erstellen einer neuen Organisationsdatenbank, da auf einfache Weise alle Personen auf Abteilungen verteilt werden können.
Das *rechte* Fenster zeigt eine Liste von Personen aus **externen Adreßbüchern**. Adreßbücher dienen unter Lotus Notes hauptsächlich zum Versand elektronischer Post. Daher hat ein Unternehmen, das Lotus Notes verwendet, meist eine Adreßdatenbank. Zumindest legen einzelne Benutzer private Adreßbücher an. Die darin definierten Personen können direkt[191] in den Organization-Object Modeler übernommen und zugewiesen werden. Das hat den Vorteil, daß bereits eingegebene Namen nicht aufwendig kopiert werden müssen. Personen, die innerhalb der Datenbank noch nicht existieren, werden nach Rückfrage generiert.

4.4.5 Die kontextsensitive Property Box

Alle Einstellungen und Optionen werden über eine Property - Box[192] vorgenommen. Sie ist der gleichnamigen Property - Box der Version 4 von Lotus Notes nachempfunden und zeigt kontextsensitiv die Eigenschaften des gerade aktuellen Objektes. In der folgenden

[191] nach Angabe der Adreßbuchdateien in den Optionen der Applikation
[192] property = engl. für ‘Eigenschaften’

Abbildung sind z.B. die Einstellungsmöglichkeiten für das Verhalten der gesamten Applikation zu sehen.
Da nicht alle Optionen gleichzeitig dargestellt werden können, wurde eine Art Notiz-block implementiert, mit dem einzelne Blätter sichtbar geschaltet werden können:

Abbildung 4-16 Property - Box

Nach Änderungen in der Property - Box erscheint in der Titelleiste ein Knopf[193], mit dem neue Werte direkt übertragen werden können.

Für das Verhalten der Applikation können auf diese Weise Einstellungen über das auto-matische Laden von Datenbanken gemacht werden. Weitere Einstellungen sind:
- automatische Speicherintervalle[194],
- Konsistenz - Tests beim Laden oder Speichern einer Datenbank[195],
- einige Druckoptionen[196],
- Optionen für Löschvorgänge des Reißwolfs[197],
- die Angabe von Adreßbüchern zur Übernahme von Personen oder Arbeitsgruppen[198].

Andere Objekte, deren Eigenschaften mit Hilfe der Property - Box dargestellt werden, sind z.B. die aktuelle Datenbank, ein Organigramm, eine Person, Abteilung, Arbeits-gruppe u.a..

[193] zweiter von links
[194] vgl. Kap. 4.1.1
[195] vgl. Kap. 4.5.2
[196] vgl. Kap. 4.4.3
[197] vgl. Kap. 4.5.6
[198] vgl. Kap. 4.5.4

Da die kontextsensitiven Änderungen der Property-Box manche Aktionen[199] etwas verlangsamen, kann sie durch den linken Knopf der Titelleiste inaktiv geschaltet werden. Sie wird daraufhin kleiner und reagiert vorerst nicht kontextsensitiv. Wird sie wieder aktiviert, zeigt sie Eigenschaften des aktuellen Objektes an.

Eine besondere Funktionalität ist die einfache Auswahl von Mitgliedern oder des Managers einer Abteilung oder Arbeitsgruppe:
Es können für eine Abteilung oder Arbeitsgruppe Listen aller Personen einer Datenbank angezeigt werden, in denen aktuelle Mitglieder hervorgehoben sind. Durch einfaches Selektieren oder Deselektieren kann der Benutzer darin die Zusammensetzung der Abteilung oder Gruppe ändern.

Bemerkung:
Der ‘Notizblock’ innerhalb der Property - Box ist eine universelle Weiterentwicklung der Klasse ‘Cabinet’, die in **[Liebrand1995]** vorgestellt wurde. Neu ist die Verwendung von Reitern mit unterschiedlicher Breite[200], die Verwendung von Sinnbildern und eine verbesserte Reaktionsgeschwindigkeit und Benutzerführung beim Blättern.

4.4.6 Interaktives Löschen - Der Reißwolf

Zum Abschluß soll noch der graphische ‘Reißwolf’ oder Mülleimer beschrieben werden. Er ist aus der Idee entstanden, daß auch Löschvorgänge interaktiv, d.h. mit Mausbewegungen möglich sein sollten:
Die Benutzerführung des Organization-Object Modeler ist hauptsächlich dafür ausgelegt, Objekte durch Mausverschiebungen in Beziehung zu setzen. Dazu existieren mehr als 90 mögliche Drag&Drop Kombinationen. Bei normalen Löschvorgängen müssen Objekte erst markiert, dann durch Tastatureingabe, Menübefehl oder einen Knopf der Werkzeugleiste gelöscht werden. Diese zwei Arbeitsschritte werden durch die ‘Mülleimerfunktion’ zu einem vereint, indem der Benutzer Objekte per Mausbewegung auf das Reißwolfbild zieht und sie durch Fallenlassen ‘entsorgt’.

[199] z.B. beim Durchblättern einer Browser - Tabelle
[200] abhängig von Wortlängen

Abbildung 4-17 Der graphische Reißwolf

4.5 Erweiterungsmöglichkeiten und bekannte Schwächen der Applikation

Zum Abschluß des Kapitels sollen die Folgen einiger konzeptioneller Aspekte beurteilt werden. Eine Beschreibung der wenigen bekannten technischen Fehler ist im Anhang aufgelistet.[201]
Einschränkungen der Applikation deuten sich hauptsächlich durch das Laufzeitverhalten bei großen Datenbanken und im Zusammenspiel mit verschiedenen Versionen von Lotus Notes an. Sie werden unten beschrieben.

Einige Kompatibilitätsprobleme ergeben sich aus der Cross - Platform Entwicklung unter StarView. So versucht die Applikation, sich den Gegebenheiten des aktuellen Betriebssystems anzupassen. Zwar sind Elemente mit einer gewünschten Schriftart und einem gewünschten Aussehen implementiert, jedoch ist nicht sicherzustellen, daß dieses Aussehen auf allen Systemen erreicht wird. Daher haben z.B. die verwendeten Dialogfenster unter OS/2 und Windows abweichende Größen. Bei der Implementierung mußte deshalb versucht werden, eine für möglichst viele Plattformen verwendbare Darstellung zu schaffen. Auf einigen Systemen können Unzulänglichkeiten in der Darstellung von Editierfeldern, Schriftarten und Schriftgrößen auftreten.

[201] vgl. Kap. A.4

Zum Laufzeitverhalten:
Der Organization-Object Modeler bildet einen Aufsatz auf eine Notes - Datenbankumgebung. Dafür stellt er besondere Funktionen und Datenstrukturen zur graphischen Darstellung und interaktiven Bearbeitung von Aufbauorganisationen zur Verfügung. Die interne Datenstruktur ist somit komplexer als die originären Datensätze der Datenbank. Daher müssen beim Ändern eines organisatorischen Objektes viele Beziehungen zu anderen Objekten beachtet und berechnet werden. Mit wachsender Anzahl unterschiedlicher Objekte summieren sich Berechnungen von Interdependenzen auf.

Zur Untersuchung des Laufzeitverhaltens wurde deshalb eine große Organisationsdatenbank erzeugt. Sie bestand aus
- 200 Abteilungen auf 13 Hierarchieebenen,
- 150 Arbeitsgruppen,
- 100 Rollen und
- 200 Personen.

Sie besaß eine Größe von 1.232 Kilobyte und sehr komplexe Strukturen in der Abteilungshierarchie und in den Arbeitsgruppenstrukturen. Ein Resultat der Arbeit mit dieser Datenbank war, daß sie auf einem PC mit folgenden Eigenschaften komfortabel bearbeitet werden konnte:
80486 CPU mit 100 MHz Taktgeschwindigkeit,
16 MB Arbeitsspeicher,
OS/2 oder Windows95 Betriebssystem.

Es hat sich allerdings eine Einschränkung bei der Eingabe neuer Elemente ergeben:
Eine Dateneingabe über Tastatur erfolgt innerhalb einer Browser - Tabelle. Diese Tabellen werden nach jeder kompletten Eingabe eines neuen Elementes neu sortiert. Mit wachsender Anzahl unterschiedlicher Organisationselemente in der Datenbank wird der Aufbau einer Tabelle langsamer. So ist eine Neueingabe in Tabellen nur bis zu einer Größenordnung von ca. 500 organisatorischen Einheiten komfortabel. Diese Beschränkung kann umgangen werden, wenn die Eingabe der Personen als letztes erfolgt oder Eingaben direkt in Lotus Notes vorgenommen werden. Bis zu der genannten Größenordnung erscheint die Eingabe innerhalb der Applikation schneller als unter Notes.

Die oben beschriebene Größenordnung einer Organisationsdatenbank dürfte jedoch im Umfeld normaler Bürokommunikation nicht erreicht werden. Deshalb wurde im Organization-Object Modeler auf eine spezielle Eingabefunktion verzichtet. Sollte die Bearbeitung größerer Datenbanken erforderlich werden, so könnte beispielsweise die Sortierung innerhalb der Browser vorübergehend abgeschaltet werden.

Interessant ist ein Performanz - Vergleich der einzelnen Versionen. Einen nicht-repräsentativen Vergleich zeigt die folgende Graphik. In ihr stehen die beim Laden und beim Konsistenztest der oben beschriebenen Datenbank gemessenen Zeiten :

Abbildung 4-18 Performanz der Applikation beim Laden und Verifizieren einer großen Organisationsdatenbank

Konsistenztests laufen unter OS/2 sehr schnell ab. Das läßt sich vermutlich darauf zurückführen, daß der Organization-Object Modeler dort als 32 - Bit Applikation implementiert ist. Dahingegen ist die OS/2 - Version beim Laden und beim Fensteraufbau spürbar langsamer. Eine mögliche Erklärung könnte sein, daß StarView die MDI - Technologie unter OS/2 anscheinend aufwendig simuliert. Interessant ist auch, daß die gleiche Windows 3.1- Umgebung unter OS/2[202] etwas schneller arbeitet.
Daher läßt sich vermuten, daß eine reine **32 - Bit Version** des Organization-Object Modelers für **Windows95** eine lohnenswerte Erweiterung der jetzigen Versionen sein könnte. Dafür müßte jedoch eine entsprechende Version der StarView Bibliotheken verfügbar sein. Diese ist für September 1996 angekündigt.

Umgang mit unterschiedlichen Notes -- Versionen:
Momentan werden unterschiedliche Versionen des Produktes Notes verwendet. Im allgemeinen kann der Organization-Object Modeler mit allen Versionen zusammenarbeiten, sofern eine passende Macroware - DLL benutzt wird.[203]
Datenbanken der Versionen 3 und 4 sind jedoch nicht in allen Fällen kompatibel. So kann z.B. die Notes Version 3 anscheinend keine Datenbanken lesen, die mit der Version 4 kompaktiert wurden. Der Organization-Object Modeler kann deshalb u.U. manche Datenbanken nicht lesen. Er meldet hierauf eine ungültige Datenbankversion mit einem Hinweis auf das Problem der Inkompatibilität. Dann bleibt dem Benutzer nur noch die

[202] als WinOS2
[203] vgl. Installationsbeschreibung im Anhang Kap. A.2

Möglichkeit, eine passende Version von Notes zu installieren und als erstes in den Systempfad zu legen.
Der Organization-Object Modeler kann keine eigenständige Konvertierung vornehmen. Das ist in einer Notes Umgebung auch nicht sinnvoll, da andere Applikationen die Datenbank nach einer Konvertierung evtl. nicht mehr lesen können.

Desweiteren sind mit dem Versionswechsel einige kosmetische Änderungen der Benutzerschnittstelle von Notes vorgenommen worden. Der Organization-Object Modeler verwendet außer der Property-Box[204] einige Dialogfenster, die der Version 3 nachempfunden sind. Um ein konformes Erscheinungsbild zu erreichen, müßten einige Fenster und Sinnbilder abgeändert werden.

[204] Notes - Version 4

Kapitel 5: Ausblick und Zusammenfassung

5.1 Ausblick

Diese Arbeit will mit dem entwickelten Organisationsmodell und der darauf aufbauenden Applikation einen Grundstein legen, der möglichst universell ausgebaut werden kann. So soll besonders das Organisationsmodell keineswegs als fixes Konzept verstanden werden. Es ist unter der Prämisse entstanden, eine generische Ausgangsbasis zu schaffen, die aber gleichzeitig einen guten Praxisbezug schafft, damit sie im bestehenden Umfeld einer Produktpalette eingesetzt werden kann.

Damit ist die Frage nach sinnvollen Weiterentwicklungsmöglichkeiten etwas schwierig zu beantworten. Man kann an dieser Stelle Entwicklungen vorschlagen, die aber in Abstimmung mit anderen wissenschaftlichen Fachgebieten z.B. der Arbeitswissenschaft und Soziologie überdacht und besonders von Erfordernissen und Wünschen der Praxis abhängig gemacht werden sollten. Mit dem GroupFlow - System ist eine effiziente Möglichkeit gegeben, das jetzige Modell in der Praxis vorzustellen oder sogar einzusetzen. Dadurch kann marktgerecht orientiert und praxisbezogen weiterentwickelt werden.

Mögliche Erweiterungen bestehen im konzeptionellen Bereich organisatorischer Strukturen, aber auch in der Integration der prototypischen Umsetzung mit anderen Werkzeugen des Workflow - Managements:

Um auch wahlweise speziellere Hierarchieformen zu unterstützen, könnte das Organisationsmodell durch andere Organisationsformen erweitert werden. Einige sind bereits im Rahmen dieser Arbeit vorgestellt worden.[205]
Stellenschlüssel und Stellenkennziffern werden bisher lediglich in der Beschreibung von Stellen angegeben. Eine Erweiterung auf universelle Schlüssel wäre denkbar, die man bei Bedarf auch in den Abteilungsobjekten einblenden könnte.
Vorstellbar wären auch Konzepte zur engeren Integration von Arbeitsgruppen in die hierarchische Abteilungsstruktur.

Eine Weiterentwicklung des Organization-Object Modelers im Rahmen der GroupFlow - Palette könnte in einer Verschmelzung der Organisationsdatenbanken mit Notes-Adreßbüchern bestehen. Damit würde eine Aufbauorganisationsdatenbank im alltäglichen Praxisbetrieb zusätzlich zu einem Namens- und Adressverzeichnis, und ihre Einträge wären direkt für e-Mail - Konzepte geeignet. Das könnte die Verwendbarkeit der Organisationsdatenbank steigern und die Datenhaltung einer Organisation redundanter machen.
Wünschenswert wäre auch ein aggregiertes Benutzerwerkzeug, das die normalerweise nachgegliederten Prozesse der Aufbauorganisationsmodellierung, der Arbeitsablaufmodellierung, der Aufbereitung und Analyse vereint. So erzeugt der Organization-Object

[205] sog. Matrix- bzw. Mehrlinienmodelle, vgl. Kap. 3.1.3

Modeler recht aufwendige graphische Strukturen, die aber entsprechende Benutzerwerkzeuge zur Ablaufmodellierung bisher nicht nutzen können. Darüber hinaus könnte die Möglichkeit zu gleichzeitiger Modellierung von Aufbau- und Ablaufstrukturen sinnvoll sein.[206]
Ein solches Projekt scheint allerdings als recht aufwendig, und es ist fraglich, ob die geforderte Komplexität mit Methoden des Software-Engineering erreicht werden kann und zuverlässig umgesetzt würde.

5.2 Zusammenfassende Betrachtung

Abschließend soll ein Bogen auf die innovativen und fast provozierenden Thesen der Einleitung gespannt werden. Ob eine Mitarbeit aller an einer flexiblen Aufbauorganisation, ökonomischen und gleichzeitig auch sozialverträglichen Erfolg haben wird, kann vor allem unter zwei Gesichtspunkten betrachtet werden:

Der ökonomische Erfolg einer integrierten, flexiblen Aufbauorganisation im Workflow - Bereich läßt sich direkt an Ergebnissen der Ablauforganisation messen. Sie kann den Ressourceneinsatz optimieren und die Projektlaufzeiten verkürzen. Durch vernetzte Datenbearbeitung kommt das Organisationsdesign multi-user fähig zum Einsatz. Es ergeben sich Vorteile in Hinblick auf Aktualität und schnelle, verteilte Verfügbarkeit des Datenbestandes. Dabei reicht die Spannweite von einer Erleichterung von Routinearbeiten bis hin zu optimaler Entscheidungsunterstützung bei der Gestaltung von Geschäftsvorgängen der Ablaufstruktur.

Dagegen ist schwieriger abzuschätzen, ob eine flexible Aufbauorganisation optimale Zielerfüllung des gesamten Systems garantieren kann. Diese Frage scheint eng mit der Erfüllung sozialer Interessen verknüpft zu sein, weil sie einen eigenverantwortlicher Umgang seiner Mitglieder voraussetzt.

Partizipation hat derart viele Facetten, graduelle Unterschiede und qualitativ unterschiedliche Realisierungsmöglichkeiten, daß die Frage, ob sie die Organisationsform *sozialverträglich* verbessern könnte, bereits an der Realität vorbeigestellt ist. *Veränderungsprozesse finden ohnehin statt*. Es geht vielmehr darum, alle Bereiche, die durch Entwicklung und Einsatz von automatischen Bürovorgängen berührt werden, einer erweiterten Mitwirkung und Mitbestimmung zu öffnen.
Zum einen gilt es, verstärkt Verantwortung auf die Mitarbeiter zu übertragen, da sie diejenigen sind, die über das nötige Wissen und Können verfügen, um auf Ausnahmen, Sonderfälle und unvorhergesehene Ereignisse situationsgerecht und flexibel zu reagieren. Zum anderen ist es wichtig, sich mehr der Dynamik der Geschäftsprozesse und den Veränderungsprozessen in der Geschäftsabwicklung zu stellen, da viele Ereignisse und

[206] etwa bei der Gestaltung von Forschungsprojekten

Probleme nicht vorhersehbar sind und demzufolge auch nicht nach einem festen Schema aufgrund starrer Vorgaben behandelt werden sollten.

Unter diesen Gesichtspunkten ermöglicht die vorliegende Arbeit einen Schritt in die Richtung offener Organisationsstrukturen. Ein soziales System ist essentiell jedoch ein konservatives System. Alle seine Teilsysteme, also die Mitarbeiter, müssen durch klare Vorteile davon überzeugt werden, die gebotenen Möglichkeiten anzunehmen. Die in dieser Arbeit entwickelte Art der Partizipation erscheint in sozialkritische Perspektive als besonders sinnvoll und attraktiv, da die Beteiligten ihre Ideen direkt und eigenverantwortlich einbringen und sich so ihr eigenes Arbeitsumfeld selber gestalten können.

Die Möglichkeit, etwas schnell (ad hoc) und lokal ändern zu können, birgt auch Gefahren:
- Evtl. sind Änderungen vorschnell
- oder nach einiger Zeit nicht mehr nachvollziehbar.

Es ist weiterhin denkbar, daß die Mitglieder einer offenen Aufbaustruktur nicht auf ein effektives Gesamtsystem hinarbeiten, sondern einfach nur ihren Teilbereich bewußt oder unbewußt möglichst angenehm gestalten:
Jeder Benutzer entwickelt seinen eigenen Stil und seine Vorlieben. Dabei passiert es häufig, daß sich unhandliche oder umständliche Handlungssequenzen ausprägen. Aus Unkenntnis oder Bequemlichkeit findet keine weitere Reflexion der eigenen Arbeitsweise statt, nicht zuletzt weil der intendierte Handlungsabschluß erreicht wird. Es ist also fraglich, ob eine weitreichende Autonomie der einzelnen Benutzer immer zu einem optimalen Ergebnis führt.

Verringerung technischer Erschwernisse und wachsende Einsicht in die Notwendigkeit von Lernprozessen sind zwar entscheidende Voraussetzungen für den Erfolg partizipativer Gestaltung, sie stellen aber für sich allein genommen noch keine ausreichende Grundlage dar. Vielmehr sollte die Intervention von Kollegen bei der Arbeit als eine unerläßliche Quelle neuer Einsichten und Verbesserungen im Umgang mit dem Gesamtsystem unterstützt werden. Insgesamt ist es nötig und wünschenswert, daß alle Beteiligten ihre Annahmen und Erwartungen überdenken, die ihre Vorstellungen von aktiver Mitarbeit prägen, damit die unbestreitbar großen Möglichkeiten im Bereich innovativer Büroumgebungen effizient, aber auch im Sinne aller genutzt werden.

6. Abbildungsverzeichnis

7. Tabellenverzeichnis

8. Literaturverzeichnis

[Agthe1989] Agthe, K.: *Organisation der Unternehmensführung in Europa und in den USA*, in: Seidel, E.; Wagner, D. (Hrsg.): *Organisation*, Wiesbaden 1989, S. 165-175.

[AlchDem1972] Alchian, A.A.; Demsetz, H.: *Production, information costs and economic organization.* In: American Economic Review 622 1972, S. 777-795.

[Bartölke1980] Bartölke, K.: *Hierarchie.* In: Grochla, E. (Hrsg.): *Handwörterbuch der Organisation.* Stuttgart 1980, Sp. 830-837.

[BjAnd1980] BjØrn-Andersen, N.: *The human side of information processing*, Amsterdam 1980.

[Bleicher1973] Bleicher, K.: *Auf dem Wege zu systemorientierten Organisationsformen*. In: Zeitschrift für Organisation 1/1973, S. 55-62.

[Bleicher1990] Bleicher, K.: *Ganzheitliches Denken in arbeitsteiligen Organisationen*. In: Bleicher, K.; Gomez, P. (Hrsg.): *Zukunftsperspektiven der Organisation*, Bern 1990, S. 11-27.

[Blum1980] Blum, Egon: *Möglichkeiten und Grenzen des Organigramms*. In: Zeitschrift für Organisation, Ausgaben 1+2, S. 42-51 u. 84-91, 1980.

[Booch1991] Booch, G.: *Object Oriented Design with Applications*, The Benjamin/Cummings Publishing Company Inc., Kalifornien 1991.

[Brings1976] Brings, K.: *Erfahrungen mit der Matrixorganisation*. In: Zeitschrift für Organisation 2/1976, S. 72-80.

[Brown1984] Brown, P.J.: *UNIX*, Addison - Wesley Publishers Limited 1984.

[BudFloydKSlaw1984] Budde, Reinhard; Floyd, Christiane; Keil-Slawik, Reinhard: *Software development and reality construction*, Berlin 1990.

[Dinkelbach1989] Dinkelbach, Werner: *Die Gestaltung der Aufbauorganisation des betrieblichen Informations- und Kommunikationssystems*, Verlag Josef Eul, Bergisch Gladbach 1989.

[Duden1989] Duden Informatik, korrigierter Nachdruck, Dudenverlag Mannheim/Wien/Zürich 1989.

[Duden1993] Duden Informatik, 2. vollst. überarb. und erw. Auflage, Dudenverlag Mannheim/Leipzig/Wien/Zürich 1993.

[Dye1983] Dye, R.: *Communication and Post-Decision Making*. In: Journal of Accounting Research 21, S. 514-533, 1983.

[Ebers1993] Ebers, M.; Gotsch, W.: *Institutionenökonomische Theorien der Organisation.* In: Kieser, A. (Hrsg.): *Organisationstheorien*, Stuttgart/Berlin/Köln 1993, S. 193-242, 1993.

[EG1982] EG: *Entschließung zum Schutz der Rechte des Einzelnen angesichts der fortschreitenden technischen Entwicklung auf dem Gebiet der Datenverarbeitung* vom 9.3.1982: Amtsblatt der EG, C87/39 vom 5.4.82.

[Fahrwinkel1995] Fahrwinkel, Uta: *Methode zur Modellierung und Analyse von Geschäftsprozessen zur Unterstützung des Business Process Reengineering*, 1. Auflage, HNI - Verlagsschriftenreihe Paderborn 1995.

[Fischeru.a.1994] Fischer, J.; Herold, W.; Dangelmaier, W.; Nastansky, L.; Wolff, R.: *Bausteine der Wirtschaftsinformatik. Grundlagen, Anwendungen, PC-Praxis*, Steuer u. Wirtschaftsverlag, Hamburg 1994.

[Frese1974] Frese, Erich: *Kontrolle als Instrument der Unternehmensführung*. In: Grochla, E. (Hrsg.): *Management*, Düsseldorf/Wien 1974.

[Frese1976] Frese, Erich: *Aufbauorganisation*, Verlag Dr. Götz Schmidt, Gießen 1976.

[Frese1987] Frese, Erich: *Unternehmensführung*, Landsberg a. Lech 1987.

[Fuchs1975] Fuchs, R.: *Hierarchie im Wandel.* In: Zeitschrift für Organisation, 1/1975, S. 9-19.

[Funk1992] Funk, W.: *Organisatorische Implikationen einer computerintegrierten Produktion (CIM). Ansatzpunkte zur Gestaltung der Aufbauorganisation, Ablauforganisation sowie der Projektorganisation bei CIM.* In: Zeitschrift für Führung und Organisation, 61. Jg., Heft 6 / 1992, S. 355-361.

[Gorz1989] Gorz, André: *Kritik der ökonomischen Vernunft, Sinnfragen am Ende der Arbeitsgesellschaft,* Berlin 1989.

[Grochla1975] Grochla, E.: *Organisationstheorie,* 2. Teilband, Stuttgart 1975.

[Habermas1981] Habermas, Jürgen: *Theorie des kommunikativen Handelns,* Band 1 u. 2, Suhrkamp Verlag, Frankfurt/M. 1981.

[HamCham1993] Hammer, Michael.; Champy, J.: *Reengineering the Corporation. A Manifesto for Business Revolution.* N. Brealy Publ. Ltd., London 1993.

[Hartwig1987] Hartwig, Rüdiger: *Die Pixel-Revolution. Wie der Computer Grafik macht,* IBM Deutschland 1987.

[Hassenstein1986] Hassenstein, Bernhard: *Information und Steuerung in Organismen,* in: Folberth, O./Hackl, C. (Hrsg.): *Der Informationsbegriff in Technik und Wissenschaft,* München/Wien 1986, S. 161 ff..

[Heindörfer1993] Heindörfer, Ralf: *Entwicklung generischer Zugangsfunktionen zu Notes-DBs in Form einer DLL auf Basis des Notes-API.* In: Business Computing & Economics, Universität Paderborn, Fachbereich Wirtschaftswissenschaften 1993.

[Heinz1995] Heinz, Oliver: *Enterprise Information Management Model as Basis for Groupware-based Workflow Management,* Studienarbeit an der Universität Paderborn, Fachbereich WI 2, Oktober 1995.

[HillFehlUl1989] Hill, W.; Fehlbaum, R.; Ulrich, P.: *Organisationslehre,* 4. Auflage 1989 (2 Bände).

[Hoppen1992] Hoppen, Dirk: *Organisation und Informationstechnologie,* Verlag Dr. Kovac, Hamburg 1992.

[IBM1989] IBM: *Systems Application Architecture, Common User Access, Advanced Interface Design Guide,* IBM Press 1989.

[IBM1991] IBM: *Object-Oriented Interface Design: IBM Common User Access (CUA) Guidelines,* IBM Press SC34-4399, 1991.

[Jaggi1969] Jaggi, B.L.: *Das Stabsproblem in der Unternehmung, Gestaltung der Stäbe und ihre Funktion in der Unternehmenspraxis,* Berlin 1969.

[KeiSl1990] Keil-Slawik, Reinhard: *Konstruktives Design: Ein ökologischer Ansatz zur Gestaltung interaktiver Systeme,* Habilitationsschrift an der TU Berlin 1990(?).

[KeiSl1991] Keil-Slawik, Reinhard: *Konstruktives Design: Ein ökologischer Ansatz zur Gestaltung interaktiver Systeme.* In: Arbeitsunterlagen zur Lehrveranstaltung ‘Gestaltung interaktiver Systeme’, TU Berlin WS 1991/1992 (Kurzfassung von **[KeiSl1990]**).

[KeiSl992] Keil-Slawik, Reinhard: *Artefakte als externes Gedächtnis. Ein ökologischer Ansatz zur Entwicklung und Bewertung von DV-Systemen.* Entwurf für einen Artikel in der Zeitschrift LOGIN 1992.

[KiesKub1983] Kieser, Alfred; Kubicek, H.: *Organisation,* 2. Auflage, Berlin 1983.

[KinLamSte1990] Kingman, L.C.; Lambert, R.E.; Steen, R.P.: *Operational image systems: A new opportunity,* in: IBM Systems Journal Vol. 29, No. 3, 1990.

[Klunder1988] Klunder, Doug: *Naming Conventions (Hungarian),* überarbeiteter Aufsatz Januar 1988, ohne Verlag.

[Knuth1988] Knuth, Donald E.: *The Art of Computer Programming,* Vol. 3: *Sorting and Searching,* Addison-Wesley Publishing Company 1988.

[KöHöKrä1975] Köhler, Joachim; Höwelmann, Rolf; Krämer, Hardt: *Analytische Geometrie und Abbildungsgeometrie*, 6. Auflage, Diesterweg Verlag, Frankfurt 1975.

[KöHöKrä1979] Köhler, Joachim; Höwelmann, Rolf; Krämer, Hardt: *Vektorielle Abbildungsgeometrie*, 2. Auflage, Diesterweg Verlag, Frankfurt 1979.

[Kosiol1976] Kosiol, Erich: *Organisation der Unternehmung*, 2. Auflage, Wiesbaden 1976.

[Kötter1993] Kötter, Wolfgang: *Auf dem Wege zur ÜBERMORGEN AG*, in: *CIM - Herausforderung an Mensch, Technik, Organisation*, Günther Cyranek und Eberhard Ulich (Hrsg.), Teubner Verlag Stuttgart 1993.

[Krallmann1989] Krallmann, H: Geleitwort in: Ulrich Busch, *Produktivitätsanalyse: Wege zur Steigerung der Produktivität*, Erich Schmidt Verlag 1989, 2. Auflage.

[Krüger1985] Krüger, Wilfried: *Bedeutung und Formen der Hierarchie*. In: Die Betriebswirtschaft, 45. Jg. 1985, S. 292-307.

[Kubicek1992] Kubicek, Herbert: *Informationstechnologie und Organisationsstruktur*, in: Frese, Erich (Hrsg.) *Handwörterbuch der Organisation*, Stuttgart 1992, S. 937 ff..

[Lauterburg1980] Lauterburg, Ch.: *Vor dem Ende der Hierarchie*, Düsseldorf/Wien 1980.

[LauxLier1987] Laux, Helmut; Liermann, Felix: *Grundformen der Koordination in der Unternehmung: Die Tendenz der zur Hierarchie*. In: Zeitschrift für Betriebswirtschaft (9/1987).

[Liebrand1995] Liebrand, Mark: *Information Object Modeler - A CASE Tool for Database Design with Lotus Notes*, Diplomarbeit an der Universität Paderborn, Mai 1995.

[Lippman1990] Lippman, Stanley: *C++, Einführung und Leitfaden*, Addison-Wesley (Deutschland) 1990.

[MedWinFlores1992] Medina-Mora, Raúl; Winograd, Terry; Flores, Rodrigo; Flores, Fernando: *The Action Workflow Approach to Workflow Management Technology*. In: CSCW 92 Proceedings 1992.

[Meyer1995] Meyer, Stefan: *Konzeption und Implementierung eines Werkzeugs zum Entwurf und zur Analyse von Bürovorgängen mit offenen objekt-orientierten Strukturen (Workflow Object Modeler)*, Diplomarbeit, Universität Paderborn, August 1995.

[Microsoft1992] Microsoft: *C/C++ Run-Time Library Reference*, 2. Auflage, Microsoft Press Redmond Washington 1992.

[Mintzberg1979] Mintzberg, H.: *The structuring of organizations*, Englewood Cliffs, New York 1979.

[Müller1995] Müller, Mirko: *Wise Man's OS/2 Warp V. 3*, Ullstein - Sybex 1995.

[NastHilp1993] Nastansky, Ludwig; Hilpert, Wolfgang: *Kritische Erfolgsfaktoren für Vorgangsbearbeitung und Workflowmanagement als Kernkomponenten in der bankbetrieblichen Dienstleistungserstellung*. In: Tagungsband zur WKWI-Konferenz Nürnberg, 7./8. Oktober 1993.

[NastHilp1994a] Nastansky, Ludwig; Hilpert, Wolfgang: *The GroupFlow System: A Scalable Approach to Workflow Management between Cooperation and Automation*, Arbeitspapier, Universität Paderborn, Februar 1994.

[NastHilp1994b] Nastansky, Ludwig; Hilpert, Wolfgang: *The GroupFlow Framework: Enterprise Model and Architecture of the Workflow System*, CSDS White Paper, Universität Paderborn, Februar 1994.

[OrfHark1992] Orfali, Robert; Harkey, Dan: *Client/Server Programming with OS/2*, 2. Auflage, Van Nostrand Reinhold, New York 1992.

[Ott1994] Ott, Marcus: *Conceptional design and implementation of a graphical workflow modeling editor in the context of distributed groupware databases*, Diplomarbeit, Universität Paderborn, Mai 1994.

[Pasch1991] Pasch, Jürgen: *Dialogischer Software-Entwurf*, Dissertation an der TU Berlin, November 1991.

[Pavone1995] Pavone Informationssysteme GmbH: *Produktinformation zu PAVONE GroupFlow und GroupProject*, Paderborn 1994-1995.

[Picot1984] Picot, Arnold: *Organisation*, in: Vahlens Kompendium der Betriebswirtschaftslehre, Band 2, München 1984, S. 95-158, 3. Auflage.

[Pieper1986] Pieper, A.: *Produktivkraft Information*, Köln 1986.

[PlesBart1990] Plesums, C.A.; Bartels, R.W.: *Large-scale document systems: an U - S - A - A. case study*. In: IBM Systems Journal Vol. 29, No. 2, 1990 (USAA ist eine amerik. Versicherungsgesellschaft).

[Qvortrup1991] Qvortrup, Lars: *Partizipative sozialorientierte Experimente mit der Informationstechnologie in Dänemark*. In: Kubicek, Herbert (Hrsg.): *Telekommunikation und Gesellschaft. Kritisches Jahrbuch der Telekommunikation*, Karlsruhe 1991, S. 78 ff..

[Raeithel1991] Raeithel, Arne: *Ein kulturhistorischer Blick auf rechnergestützte Arbeit*. In: *Sichtweisen der Informatik*, Vieweg Verlag 1991.

[Röthig1989] Röthig, Peter: *Perspektiven für Organisation und Führung von Unternehmen.* In: **[SeiWag 1989]** S. 311-323.

[Schertler1988] Schertler, Walter: *Unternehmensorganisation: Lehrbuch d. Organisation u. strateg. Unternehmensführung*, 3. erw. Auflage, Oldenbourg 1988.

[Schneider1985] Schneider, D.: *Allgemeine Betriebswirtschaftslehre*, München 1985.

[Schüler1990] Schüler, W.: *Informationstechnologie und organisatorischer Wandel: Über die Grundlagen der Interdependenz*, Diskussionsarbeit Nr. 213, Universität Bielefeld, Fachbereich Wirtschaftswissenschaften 1990.

[SeiWag1989] Seidel, Eberhard; Wagner, Dieter (Hrsg.): *Organisation*, Gabler, Wiesbaden 1989.

[Staehle1991] Staehle, Wolfgang H.: *Management, Eine verhaltenswissenschaftliche Perspektive*, Verlag Franz Vahlen, München 6. Auflage 1991.

[Staerkle1989] Staerkle, Robert: *Organisationskonzeptionen erfolgreicher mittlerer Industrieunternehmen.* In: **[SeiWag 1989]** S. 151-163.

[StarView1992a] Star Division: StarView C++ Klassenbibliothek: *Benutzerhandbuch* der Version 2.0, Star Division, Hamburg 1992.

[StarView1992b] Star Division: StarView C++ Klassenbibliothek: *Programmierhandbuch* der Version 2.0, Star Division, Hamburg 1992.

[StarView1992c] Star Division: Online Hilfe zum StarView Browser, Star Division, Hamburg 1992.

[Steinmüller1993] Steinmüller, Wilhelm: *Informationstechnologie und Gesellschaft: Einführung in die Angewandte Informatik*, Darmstadt: Wiss. Buchges. 1993.

[Stroustrup1989] Stroustrup, Bjarne: *The C++ Programming Language*, Addison-Wesley Publishing Company 1989.

[Tannenbaumu.a.1974] Tannenbaum, A.S.; Kavcic, B.; Rosner, M.; Vianello, M.; Wieser, G.: *Hierarchy in Organizations. An international Comparison*. San Francisco/Washington/London 1974.

[Tarjan1983] Tarjan, B.: *Data Structures and Efficient Algorithms*, Bände 2 u. 3, SIAM Monograph 1983.

[Voet1987] Voet, Ludwig: *Organisation durch vernetzte Bürosysteme. Wohin verabschiedet die Informationstechnik den Taylorismus?* In: Paul, M.: *Computerintegrierter Arbeitsplatz im Büro*, 17. Jahrestagung der Gesellschaft für Informatik, Berlin/Bonn 1987, S. 924 ff..

[WainFran1984] Wainwright, Judith; Francis, Arthur: *Office automation, organisation and the nature of work*, Gower Publishing Comp. Ltd., Hampshire 1984.

[Weidner1977] Weidner, Walter: *Organisation in der Unternehmung: Aufbau- und Ablauforganisation*, Stollfuß Verlag, Bonn 1977.

[Wild1973] Wild, Jürgen: *Hierarchie in der Krise?* In: Zeitschrift für Organisation 1/1973, S. 45-54.

[WinRiehm1985] Wingert, Bernd; Riehm, Ulrich: *Computer als Werkzeug. Anmerkungen zu einem Mißverständnis.* In: Technik und Gesellschaft, Jahrbuch 3, S. 107-131, Campus Verlag 1985.

[Wirth1986] Wirth, Niklaus: *Algorithmen und Datenstrukturen*, B.G.Teubner, Stuttgart 1986.

[ZünGrunt1980] Zündorf, L.; Grunt, M.: *Hierarchie in Wirtschaftsunternehmen*, Frankfurt/M. 1980.

ANHANG

A.1 Systemvoraussetzungen

Die beschriebenen Konzepte wurden im Organization-Object Modeler realisiert. Die Prototyp ist mit einer graphischen Benutzerschnittstelle ausgestattet. Im Grunde ist er ein graphischer Aufsatz zur Visualisierung, Verwaltung und Organisation der Strukturen innerhalb der zugrundeliegenden Organisationsdatenbanken.
Der Bearbeitungskomfort wächst mit der Anzahl der gleichzeitig darstellbaren Objekte, da der graphische und logische Zusammenhang der Strukturen transparenter wird. Deshalb ist für eine konzeptionelle, gestalterische Arbeit ein möglichst großer Bildschirm mit entsprechender Auflösung empfehlenswert. Dazu gehört eine entsprechend ausgestattete Hardware für einen ausreichend schnellen Bildaufbau. Außerdem wächst die Geschwindigkeit, wenn alle Teile der Applikation und des Betriebssystems gleichzeitig im Hauptspeicher gehalten werden.

Grundsätzlich kann jedoch jeder Computer für lokale Anpassungen verwendet werden, auf dem MS-Windows und LOTUS Notes installiert ist. So reicht z.B. ein Notebook mit dreifarbiger Hercules - Graphik und einem 80286 Prozessor. Das bedeutet, daß auch ein Aussendienstmitarbeiter mit relativ minderwertiger Ausstattung die Applikation benutzen kann.

Für die verschiedenen Betriebssysteme können die Werte aus Tabelle A 1 als Richtlinie gelten. Auf der rechten Spalte der Tabelle sind die Anforderungen an einen Design-Arbeitsplatz aufgeführt, während die mittlere Spalte die Anforderungen zeigt, die für ein Arbeiten ohne größere Verzögerungen nötig ist. Die mit '*' gekennzeichneten Werte sind der StarView Dokumentation entnommen (StarView Benutzerhandbuch Kapitel 1.1 *Hard- und Software Voraussetzungen*, Seite 2f). Alle anderen Werte haben sich während der Umsetzung der Programmstrukturen herauskristallisiert und sind subjektive Kriterien des Autors:

Voraussetzung:	Minimum	Empfehlenswert
Prozessor	IBM PC mit 80386 Prozessor*	80486 Prozessor mit mind. 66 MHz
Graphik	alle Grafikadapter, die MS - Windows bzw. OS/2 unter-stützt, allerdings: bei mono-chromer Darstellung Informati-onseinbußen	VGA mit mind. 1024 x 768 Bild-punkten und entsprechendem Bildschirm, um die Möglichkeiten der graphischen Präsentation voll ausnutzen zu können
Hauptspeicher	**Windows:** mind. 4 MB freien Hauptspeicher*, da das Betriebssystem und eine Hintergrundversion von Lotus NOTES laufen muß **Windows NT:** 16 MB* **OS/2:** 8 MB*	**Windows:** mehr als 8 MB **Windows NT:** 16 MB* **OS/2:** mehr als 8 MB (siehe Anmerkung 1 unten)
Maus	ja	ja
Betriebssystem	**Windows:** MS-DOS 3.x oder höher* Windows 3.1 oder höher* **Windows NT:** Windows NT 3.1 oder höher* **OS/2:** OS/2 2.1 oder höher*	**Windows:** Windows 95 in möglichst aktueller, korrigierter Version **Windows NT:** bisher keine Vergleichsmöglichkeit, deshalb Empfehlung lt. StarView **OS/2:** OS/2 Warp 3

A.2 Installation

Für die in dieser Dokumentation dargelegten Konzepte wurde eine prototypische Applikation entwickelt, der 'Organization-Object Modeler'. Er ist für die Betriebssysteme OS/2 und MS - Windows jeweils in deutscher und englischer Version vorhanden. Die entsprechenden Dateien sind in der Lotus NOTES - Version dieses Dokumentes im Anhang angebunden.

A.2.1 Installation unter MS - Windows

Die Installation unter MS - Windows ist für die verschiedenen Versionen
- Windows 3.x
- Windows for Workgroups 3.x
- Windows95
- Windows NT 3.x
nahezu gleich.

Vorgehensweise:
1. Es wird ein Applikationsverzeichnis erstellt, z.B. c:\gf-orga
2. folgende Dateien werden in das Verzeichnis kopiert:
    - gf-orga.exe in englischer oder deutscher Version
    - gf-orga.ini

Folgende Dateien müssen entweder im Systempfad liegen oder werden ebenfalls in das Applikationsverzeichnis kopiert:

- SV221MW.DLL
- TL221MW.DLL
- macrware.dll in einer entsprechenden Version.

Anmerkungen:

- Die Macroware - DLL existiert in verschiedenen Versionen für Lotus NOTES. Für die 32 Bit Version von Lotus NOTES muß eine spezielle Macroware - DLL verwendet werden.
- Die Datei gf-orga.ini ist nicht notwendig und wird bei Bedarf automatisch mit Standardwerten erstellt.
- Z.Zt. können nicht alle Produkte, mit denen der Organization Modeler zusammen arbeitet, verkettete Arbeitsgruppen anzeigen und verarbeiten. Deshalb steht in der gf-orga.ini Datei ein Kompatibilitäts - Eintrag der das Ineinanderschachteln von Arbeitsgruppen verhindert:

  ```
  [Compatibility]
  AllowWorkgroupInWorkgroup=No
  ```

  Zu Vorführzwecken kann dieser Eintrag gelöscht oder auf 'yes' gesetzt werden.

A.2.2 Installation unter OS/2 jedoch als Windows Programm

Die Installation erfolgt, wie unter A.2.1 beschrieben.

A.2.3 Installation unter OS/2

Vorgehensweise:
1. Es wird ein Applikationsverzeichnis erstellt, z.B. c:\gf-orga
2. folgende Dateien werden in das Verzeichnis kopiert:
   - gf-orga.exe für OS/2 in englischer oder deutscher Version
   - gf-orga.ini

Folgende Dateien müssen entweder im Systempfad liegen oder werden in das Applikationsverzeichnis kopiert:
   - SV221CO.DLL
   - TL221CO.DLL
   - $mcrware.dll.

Anmerkungen:
- Wie unter MS-Windows ist Datei gf-orga.ini nicht notwendig. Sie wird bei Bedarf automatisch mit Standardwerten erstellt.
- Z.Zt. können nicht alle Produkte, mit denen der Organization Modeler zusammenarbeitet verkettete Arbeitsgruppen anzeigen und verarbeiten. Deshalb steht in der gf-orga.ini Datei ein Kompatibilitäts - Eintrag der das Ineinanderschachteln von Arbeitsgruppen verhindert:

  ```
  [Compatibility]
  AllowWorkgroupInWorkgroup=No
  ```

  Zu Vorführzwecken kann dieser Eintrag gelöscht oder auf 'yes' gesetzt werden.

A.3 Übersetzen, Debugging

A.3.1 Übersetzen des Quellcodes

Die Quelldateien werden für die Betriebssysteme MS-Windows oder OS/2 übersetzt. Der Übersetzungsvorgang besteht jeweils aus vier Stufen:

Stufe	Vorgang
1	Übersetzung der StarView - Ressourcedatei 'gf-orga.src' mit Hilfe des StarView Ressource - Compilers in eine betriebssystemabhängige Datei 'gf-orga.res'
2	Übersetzen der C++ Programmdateien
3	Binden der Objektdateien und Bibliotheken zur Programmdatei 'gf-orga.exe'
4	Anbinden der Ressource - Datei 'gf-orga.res' an die Programmdatei

Für diese Vorgehensweise werden verschiedene Projekt - Dateien verwendet.
Für die nachfolgend angeführten Beispiele wird angenommen, daß sich die Quelldateien in einem Verzeichnis '\gf-orga' befinden.

A.3.1.1 Übersetzung unter Windows

Es müssen folgende Programme installiert sein:
- StarView für Visual C++
- Visual C++ Version 1.5 oder höher

Stufe 1:
Achtung: der StarView Ressource-Compiler benötigt ca. 550 KB freien Hauptspeicher. Wenn nicht korrekt übersetzt wird, liegt es in den meisten Fällen an zu kleinem Arbeitsspeicher.

StarView kann betriebssystemunabhängig nur Bitmaps verwalten. Sinnbilder (engl.: icons) und Mauszeiger (engl.: pointer) liegen in einem systemabhängigen Format und werden bei Bedarf übersetzt. Besonders die Qualität der Mauszeiger leidet etwas darunter.

Zum Übersetzen der StarView - Ressourcedatei wird am einfachsten die Batch-Datei (Stapeldatei) 'svresmak.bat' in das StarView-Verzeichnis 'sv \ bin \' kopiert. Sie ermöglicht schnelles Übersetzen der StarView-Ressource aus der graphischen Benutzeroberfläche von Visual C++ heraus. Die Datei wird in den Menüpunkt 'Tools' aufgenommen, indem folgende Einstellungen im Menüpunkt 'Optionen ->Tools' gemacht werden:

Tools

Menu Contents:

&App Studio
CodeView for &Windows
CodeView for &MS-DOS
StarView-DesignEditor
StarView-Resourcen compilie
StarView Dokumentation

Add...
OK
Delete
Cancel
Move Up
Help
Move Down

Command Line: C:\SV\BIN\SVRESMAK.BAT
Menu Text: StarView-Resourcen compilieren
Arguments: $Proj
Initial Directory: $PROJDIR

Ask for Arguments

Abbildung: "Einbindung der Stapeldatei 'svresmak.bat'
Quelle: Datenbank PAVONE Class Library, Kapitel: Technical Info - Einführung

Anmerkung:
Der Vorteil dieser Stapeldatei ist, daß auch die StarView - Ressourcen anderer Projekte übersetzt werden können, wenn sie den gleichen Programmnamen wie das Projekt haben.
Beispiel: ein Projekt 'Editor' hat die Ressourcedatei 'editor.src'.

Zur Übersetzung der StarView - Ressourcedatei ist es nötig, daß die Projektdatei 'gf-orga.mak' geöffnet ist.
Das geschieht im Menüpunkt 'Project' - 'Open Project'.
Wenn sich die Quelldateien nicht im Verzeichnis '\gf-orga' befinden, erfolgt eine Meldung, daß Pfade automatisch angepaßt werden.

Achtung:
Die Bibliothekspfade werden nicht automatisch angepaßt. Befinden sich die StarView-Bibliotheken nicht in einem Verzeichnis 'c:\sv\lib', so müssen die betreffenden Dateien im Menüpunkt 'Project' - 'Edit' neu angegeben werden.

Nach diesen Vorbereitungen wird die Ressourcedatei unter dem Menüpunkt 'Tools' - 'StarView-Resourcen compilieren' übersetzt.

Stufen 2 bis 4:
Übersetzen, Binden und Anbinden der Ressourcedatei erfolgt in einem Schritt durch Aufruf der Projektdatei.
Dies geschieht entweder in einer Kommandozeile mit dem Befehl: 'nmaker gf-mod2.mak' oder im Menüpunkt 'Project' von Visual C++.
Die fertige Programmdatei 'gf-orga.exe' steht danach im Verzeichnis '\gf-orga'.

Wechseln der aktuellen Sprache:
Die Sprache wird in der Datei 'svresmak.bat' mit der Einstellung '-lg' geändert.
-lgENGLISH erstellt eine englische Version,
-lgGERMAN erstellt eine deutsche Version,
-lgSPANISH erstellt eine spanische Version.
Spanisch ist vorbereitet, d.h. Felder für spanische Zeichenketten sind in der Ressourcedatei erstellt, aber die Zeichenketten sind bisher nicht alle übersetzt.

Anmerkungen:

- Zum Kompilieren wird unter MS-Windows ein 32-Bit DOS Extender - Programm verwendet. Deshalb darf der Befehl
  'make gf-mod2.mak' nicht verwendet werden.
- Aus dem gleichen Grunde ist es nicht möglich, das Windows - Programm unter OS/2 zu erstellen.
  OS/2 verhindert, daß der 32 Bit - DOS Extender geladen wird.
- Ob eine Debugging oder Release Version erzeugt wird, kann in der Projektdatei 'gf-orga.mak' durch den Schalter

Debug eingestellt werden. Debug = 0 erstellt eine Release Version, Debug = 1 eine debuggingfähige Version.

- Die Einstellungen über Stack- und Heap - Größe sind kritisch und sollten nur mit Vorsicht geändert werden.

A.3.1.2 Übersetzung unter OS/2

Es müssen folgende Programme installiert sein:

- Toolkit 2.1 oder höher
- Compiler C Set 2 ++
- Linker link386.exe
- StarView für C Set 2 ++

Es empfiehlt sich ein Hauptspeicher von mehr als 8 MB, da der Compiler sehr spei-
cherintensiv arbeitet. Bei 8 MB oder weniger wird häufig auf Festplattenspeicher aus-
gelagert und der Übersetzungsvorgang dauert lange.

Stufe 1:
StarView kann betriebssystemunabhängig nur Bitmaps verwalten. Sinnbilder (engl.: icons) und Mauszeiger (engl.: pointer) liegen in einem systemabhängigen Format und werden bei Bedarf übersetzt. Die Qualität der Mauszeiger leidet etwas darunter.

Stufen 1 bis 4:
Alle 4 Stufen erfolgen durch Aufruf der OS/2 - Projektdatei 'gf-orga.cst'.
Dies geschieht in einer OS/2 - Kommandozeile mit dem Befehl: 'nmake gf-orga.cst'.
Als Ergebnis steht die Programmdatei 'gf-orga.exe' im Unterverzeichnis \gf-orga\os2\.

Wechseln der aktuellen Sprache:
Die Sprache wird in der Datei 'gf-orga.cst' mit dem Parameter 'Language=' eingestellt.
Language=ENGLISH erstellt eine englische Version,
Language=GERMAN erstellt eine deutsche Version,
Language=SPANISH erstellt eine spanische Version.
Spanisch ist vorbereitet, d.h. Felder für spanische Zeichenketten sind in der Ressour-
cedatei erstellt, aber die Zeichenketten sind bisher nicht alle übersetzt.

Anmerkung:

- Alle Zwischendateien werden im Unterverzeichnis '\gf-orga\os2' erstellt, damit gleichlautende Dateien im Verzeichnis '\gf-orga' nicht überschrieben werden und bei späterem Übersetzen der Windows - Version nicht vollständig neu übersetzt werden müssen.

- Mit Hilfe der Projektdatei 'gf-orga.cst' wird eine Linkdatei gf-orga.lnk und eine De-
finitionsdatei gf-orga.def erzeugt, die das Verhalten des lauffähigen Programms be-
schreiben. Bei abgebrochenen Übersetzungsvorgängen müssen diese Dateien u.U. erst gelöscht werden, da ansonsten veraltete Versionen verwendet werden können.

A.3.2 Debugging

Der Quellcode kann auf allen verwendeten Betriebssystemen unter den dort verfügbaren Debuggern getestet werden.
In Ausnahmefällen treten Fehlermeldungen nur bei einem Betriebssystem auf, während der Code auf dem oder den anderen Betriebssystemen einwandfrei arbeitet. Für diese Kategorie von Fehlern oder für systemspezifische Code -Teile muß ein Debugger des betreffenden Systems verwendet werden.

A.3.2.1 Debugging unter Windows

Zur Verfügung steht unter Visual C++ der CodeView - Debugger in einer textorientierten DOS - Version und einer Version, die direkt auf den Fenstern der graphischen Benutzeroberfläche arbeitet. Die DOS Version arbeitet etwas zuverlässiger.

Wenn ein zweiter Bildschirm zur Verfügung steht, ist der CodeView DOS - Debugger gut einsetzbar, da dann seine Bildschirmausgaben auf den textorientierten Sekundär-Bildschirm gelegt werden können. Anderenfalls ist dem windows-orientierten Debugger der Vorzug zu geben, da er die Möglichkeiten der graphischen Benutzeroberfläche ausnutzt und direkt in den Fenstern des Quellcodes abläuft.

Anmerkung
Der CodeView Windows - Debugger arbeitet im sogenannten 'Hard Mode' stabiler, da in diesem Modus alle Tastatur- und Mauseingaben abgefangen werden.

A.3.2.2 Debugging unter OS/2

Unter OS/2 steht mit dem C Set 2 ++ Compiler nur der Standard-Debugger zur Verfügung. Er ist langsamer als die beiden Windows - Debugger. Er wird in einer OS/2 Kommandozeile mit dem Befehl 'ipmd gf-orga.exe' aufgerufen.
Debugging-fähige Dateien werden mit Änderung zweier Zeilen in der Projektdatei 'gf-orga.cst' und Neuübersetzung erzeugt. In der Projektdatei stehen jeweils zwei Zeilen für den Compiler und den Linker. Eine der Zeilen wird jeweils durch Auskommentieren ungültig gemacht. Genaue Vorgehensweise und Bedeutung wichtiger verwendeter Parameter wird innerhalb der Datei beschrieben.

Welcher Debugger ist wann vorteilhafter?

Bei den Vorgängerversionen von Windows95 ist der Debugger sehr häufig abgestürzt.
Bei jedem Absturz wurde der verwendete Speicherbereich gesperrt, so daß nach jeweils einigen Abstürzen das Betriebssystem neu geladen werden mußte. Bei Windows95 wird der Speicher freigegeben, dafür waren aber Abstürze des Debuggers häufig nicht mehr zu beheben.

Deshalb wird normalerweise wohl am effizientesten unter Windows 'debuggt', weil das entscheidend schneller geht als unter OS/2. Treten jedoch hartnäckige Fehler bei der Fensterverwaltung des Organization-Object-Modelers auf oder muß bei jedem MS-Windows Start eine Netzwerkumgebung geladen werden, kommt man mit OS/2 sicherer zum Ziel.

A.4 Bekannte technische Fehler

Bei der Arbeit mit dem Organisation Object Modeler hat sich eine Reihe von Fehlern oder Fehlerquellen ergeben. Diese sollen an dieser Stelle kurz erwähnt werden. Eine genauere Beschreibung und Anleitung zum Umgehen dieser Fehlerquellen liefert [Meyer1995] , Kapitel 5. Portierung auf verschiedene Betriebssysteme.

a) Programmumschaltung mittels Alt + Tab
Unter Windows 3.x funktioniert vom Organization-Object Modeler die Programmumschaltung zu anderen Applikationen mittels der Tastenkombination Alt + Tab nur in bestimmten Sonderfällen (z.B. wenn ein nichtmodales Dialogfenster geöffnet ist). Der Fehler liegt offensichtlich an der verwendeten StarView - Klassenbibliothek und tritt in verschiedenen Applikationen auf, die StarView verwenden.
Bei Windows95 oder OS/2 funktioniert die Programmumschaltung uneingeschränkt.

b) ältere Organisationsdatenbanken der GroupFlow Umgebung
Bei alten Organisationsdatenbanken können Ansichten fehlen oder Felder mit Schlüsselbegriffen anders benannt sein, als das der Organization-Object Modeler erwartet. Soweit alte Standards bekannt waren, werden sie berücksichtigt. Bei zu alten Prototyp-Datenbanken können aus diesen Gründen u.U. Teile nicht geladen werden. So kann es beispielsweise vorkommen, daß alle Personen, Abteilungen und Rollen eingelesen werden, aber keine Arbeitsgruppen.
Der Organization-Object Modeler weist in so einem Fall auf die fehlende Ansicht oder das fehlende Feld hin und erklärt, welche Teile nicht geladen werden können.

Alle Datenbanktypen der GroupFlow - Produkte sind getestet und müßten ohne Einschränkung verarbeitet werden können.

c) alte Versionen der Macroware DLL
Der Organization-Object Modeler testet bei Programmstart, ob alle wichtigen DLLs gestartet werden können. Wenn nicht, produziert er eine entsprechende Warnungs-

meldung. Mit einigen älteren Versionen der Macroware DLL startet der Modeler zwar, es kann aber zu Abstürzen oder Fehlfunktionen der Applikation kommen. Daher sollte sichergestellt werden, daß die vom Modeler benutzte Version nicht älter als März 1996 ist. Unter OS/2 funktioniert der Modeler offensichtlich mit allen Versionen korrekt.

d) 32 Bit - Version der Macroware DLL mit Notes 4 für Windows95
Für die Arbeit mit Notes 4 für Windows 95 ist eine 32 Bit Macroware DLL erforderlich, die auf besondere technische Details der 32 Bit - Technologie zugeschnitten ist. Mit einer 16 - Bit Macroware und der 32 Bit Version von Notes 4 arbeitet der Organization-Object Modeler nicht korrekt zusammen.

e) fehlerhafte Darstellung von Dialogfenstern unter Windows95
Die StarView - Bibliotheken wurden entwickelt, als Windows95 noch nicht auf dem Markt war. Bei der Darstellung von Dialogfenstern können im Rahmen Teilstücke unsichtbar sein und in den Dialogfenstern können an bestimmten Stellen kurze schwarze Balken auftauchen.
Diese Fehler sollten mit einer neueren StarView - Version behoben sein.

f) Abstürze beim Scrollen unter OS/2
Beim Scrollen von Graphiken unter OS/2 kann es zu Abstürzen kommen. Da der Fehler unter MS-Windows nicht auftritt, ist wahrscheinlich eine Unzulänglichkeit der StarView - Bibliotheken die Ursache, die bisher noch nicht umgangen werden konnte.

A.5 Fachbegriffe, Warenzeichen, Naming Conventions (Hungarian)

A.5.1 Verwendete Fachbegriffe

Cross Platform - Entwicklung
Entwicklung von Quellcode, der auf verschiedenen Betriebssystemen ohne Adaption lauffähig übersetzt werden kann.

DLL, Direct Link Library
Bibliothek aus Befehlen und Makrobefehlen, die ein Computerprogramm zur Laufzeit aufrufen und benutzen kann.

Drag & Drop
Bezeichnet einen Vorgang, bei dem mit gedrückter Maustaste ein Objekt der graphischen Benutzeroberfläche verschoben und beim Loslassen der Taste meist in ein anderes Fenster hineingesetzt wird.

Parallelprojektion

"Darstellungsweise räumlicher Gegenstände, bei der die Verjüngung der weiter entfernt liegenden Teile nicht berücksichtigt wird; entfernt liegende Teile gleicher Größe wirken in dieser Darstellung unnatürlich groß" **[Hartwig1987]** S. 199.

Perspektivprojektion
"Darstellungsweise räumlicher Gegenstände, bei der die Verjüngung der weiter entfernt liegenden Teile berücksichtigt wird" **[Hartwig1987]** S. 199.

Replika-ID
Eindeutige Identifikationsnummer zum Auffinden gleicher Datenbanken bei Repliziervorgängen.

Replizieren
Abgleich von Notes - Datenbanken, die die gleiche Replika-ID besitzen. Dabei werden Dokumente ausgetauscht, so daß die Datenbanken nach einer vollständigen Replikation einen identischen Inhalt haben. Es ist auch selektive Replikation möglich, die den Austausch in einer oder anderer Richtung filtert.

A.5.2 Warenzeichen

Produktname / Warenzeichen	**Firma**
80386, 80486, Pentium	Intel Corporation
MS-DOS, Microsoft, Visual C++, MS-Windows, OS/2, Presentation Manager (PM), Windows95, Windows for Workgroups	Microsoft Corporation
CUA, IBM PC, VGA	International Business Machines (IBM) Corporation
Helvetica	Linotype
Macroware, GroupFlow	PAVONE
NOTES	Lotus
StarView, StarWriter	StarDivision
APPLE, Macintosh, System 7	APPLE Computer Inc.
Motif	Open Software Foundation, Inc.
Sun, Solaris, SPARC, SPARCStation	Sun Microsystems, Inc.
X-Windows	Massachusetts Institute of Technology

A.5.3 Naming Conventions

Namenskonventionen bei der Benennung von Variablen innerhalb des Quellcodes, in Anlehnung an die von Charles Simonyi aufgestellten "Meta-Programming" Thesen (vgl.: **[Klunder88]**):

Variablen:

b	ein Byte, insbesondere benutzt für Boolesche flags
e	enum
fl	float
h	handle, meist ein Zeiger auf einen Zeiger
l	long
p	Zeiger
pv	Zeiger auf den Type void
s	short
strc	struct
sz	Zeichenkette mit abschließendem Null - Character
u	unsigned
ul	unsigned long
us	unsigned short
w	word

Konstanten:

NULL	0
TRUE	Wert ungleich 0
FALSE	0

Alle Konstanten werden in Großbuchstaben geschrieben.

Variablen von StarView - Klassen:

Vor den Klassennamen wird jeweils der Kleinbuchstabe 'a' gesetzt.
Beispiele hierfür sind:
aString = Präfix für Variablen der Klasse String (Zeichenkette)
aRectangle = Präfix für Variablen der graphischen Klasse Rectangle (Rechteck)

Anmerkung:
Mit der Namensgebung innerhalb des Quellcodes soll eindeutig der Typ der Variablen erkennbar sein. Desweiteren wurden lange, erläuternde Bezeichnungen bevorzugt, damit der insgesamt recht umfangreiche Quellcode besonders bei ungekapselten Variablen verständlich bleibt.
Das führt bisweilen zu ungewöhnlich langen Bezeichnern, wie z.B. der Variablen 'bCheckOrgaDBOnLoading'. Diese Variable kann die Wahrheitswerte TRUE und FALSE enthalten und zeigt an, ob eine Organisationsdatenbank direkt nach einem Ladevorgang auf Konsistenz geprüft werden soll.

A.6 Struktur der Quelldateien und Klassen

Dieses Kapitel gibt einen Einstieg in die Struktur des Quellcodes. Es richtet sich an Programmierer, die ein Verständnis der zugrundeliegenden Strukturen erwerben wollen, um die Applikation ganz oder teilweise zu adaptieren, zu portieren oder zu erweitern.
Es ist in den Anhang gestellt worden, da die Konzepte des Prototypen Organisation Object Modeler wohl besser ohne zu genaue Kenntnis der programmierten Strukturen erkennbar werden. Das erklärt sich durch die konsequent objekt-orientierte Implementierung, die das Zusammenspiel einzelner (z.T. graphischer) Objekte nur über Schnittstellen bzw. Nachrichtenkanäle für die Kommunikation festlegt. Somit ist eine sequentielle Ablaufstruktur nicht direkt aus den Quelldateien ersichtlich.

Insgesamt ist der Quellcode mit ca. 2,7 MB sehr umfangreich. Deshalb kann an in diesem Rahmen kein umfassender Überblick vermittelt werden. Kap. A.6.1 listet die einzelnen Dateien und deren Funktion auf. Das ist für ein erstes Installieren und Probeübersetzen interessant. Kap. A.6.2 beschreibt die Klassenhierarchien, d.h. die Struktur, mit der die implementierten Klassen untereinander abgeleitet sind.

A.6.1 Quelldateien

Es folgt eine Auflistung der einzelnen Dateien des Quellcodes, deren Größe in Byte und eine kurze Beschreibung des Inhalts oder Verwendungszwecks.

Projektdateien:

GF-ORGA . CST	22.378	Projektdateien zum Übersetzen unter OS/2
GF-ORGA . MAK MAIN. DEF	24.765 839	Visual C++ Projektdateien zum Übersetzen unter Windows
GF-ORGA . SRC GF-ORGA . HRC	246.505 10.843	StarView spezifische Ressourcedateien
HELP . HHC	5.123	Global definierte Hilfenummern für die Hilfeapplikation
SVRESMAK . BAT	204	Stapeldatei zum Übersetzen der StarView spezifischen Ressourcedateien
	gesamt: 310 KB	

C++ Quelldateien:
Die Dateien mit Endung '.hpp' beschreiben jeweils Instanzen und Schnittstellen der verschiedenen implementierten Klassen. Die Funktionen der Klassen sind in den Dateien mit Endung '.cpp' oder '.cxx' realisiert.

$MacrWare.Lib	10.752	Migrationsdatei zur Verwendung der Macroware DLL unter OS/2

_Browser.hpp _Browser.cpp	20.981 133.332	Neu-Implementierung einer StarView DLL, die sich als zu fehlerhaft erwies. Dient zur Darstellung und Bearbeitung von Informationen in Tabellen
4Browser.hpp PersoBsr.cpp RolesBsr.cpp UnitsBsr.cpp WorkgBsr.cpp	14.211 59.835 35.576 44.223 48.491	Dateien zur Darstellung und Bearbeitung von Personen, Rollen, Abteilungen und Arbeitsgruppen in tabellarischer Form
Cabinet2.hpp Cabinet2.cpp	2.634 14.241	Klasse zur Darstellung von Informationen in Notizblock-ähnlicher Form mit Hilfe von Reitern
CheckDB.hpp CheckDB.cpp	8.133 96.611	Klassen zum Überprüfen der Konsistenz einer Datenbank und zur Erstellung von Statistiken
Databses.hpp Databses.cpp	2.837 17.026	Verwaltung der geladenen Lotus Notes Datenbanken
Dialogs.hpp Dialogs.cpp	6.837 66.479	Dialogboxen zur Auswahl einer Datenbankdatei und zum Serverzugriff
Dialogs2.hpp Dialogs2.cpp	9.153 52.519	Dialogboxen zum Nachfragen bei unbekannten Objekten oder Inkonsistenzen
Dialogs3.hpp Dialogs3.cpp	4.017 14.820	Dialogboxen zum Einfügen von Inhalten der Zwischenablage
Dialogs4.hpp Dialogs4.cpp	10.379 65.745	Dialogboxen für Fehlerberichte und Statistiken
Dialogs5.hpp Dialogs5.cpp	5.089 54.610	Dialogboxen für schnelles Einfügen von Objekten (Drag&Drop Listboxen)
DragPool.hpp DragPool.cpp	3.665 5.504	Klassen zur Verwaltung der durch einen Drag&Drop Vorgang angesprochenen Objekte
FixedBtn.hpp FixedBtn.cpp	909 3.071	Mit Mausklick ansprechbare Bitmaps
GF-Orga.ini	1.740	Initialisierungsdatei der Applikation mit Einstellungen zum Laufzeitverhalten
Graphs.hpp UnitGrph.cpp WGGraph.cpp	22.374 116.130 88.089	Klassen zur Darstellung und Bearbeitung von Abteilungen und Arbeitsgruppen in graphischer Form
MacrWare.h	7.375	Header - Datei zum Einbinden der Macrware Bibliotheken
Macrware.lib	23.040	Pseudo - Bibliothek in der die Schnittstellen der Windows - Macroware DLL definiert sind
Main.hpp Main.cpp	12.126 86.648	Klassen zum Erzeugen der Applikationsfenster und der Nachrichtenkanäle, Hauptklasse der Applikation
MyBrowsr.hpp MyBrowsr.cpp	6.214 33.117	Ableitung der allgemein implementierten Tabellen aus _Browser.hppzur Benutzung in den 4 Tabellenformen in 4Browser.hpp

Datei	Größe	Beschreibung
MyClipbd.hpp MyClipbd.cpp	3.821 21.720	Klasse für die Applikations-interne Zwischenablage
OrgaDB.hpp OrgaDB.cpp	34.051 177.841	Klasse zum Verwalten einer Organisationsdatenbank
OrgaMDI.hpp OrgaMDI.cpp	4.666 39.094	Klasse zur Darstellung des Hauptfensters einer Organisationsdatenbank
Peanuts.hpp Peanuts.cpp	3.830 32.939	kleine Funktionen, die von sehr vielen Klassen aufgerufen werden, meist zur Listenbearbeitung
PersCtrl.hpp PersCtrl.cpp	3.888 16.128	Klasse zur graphischen Darstellung einer Person innerhalb einer Arbeitsgruppe
Person.cpp	99.785	Verwaltung der Daten einer Person, Schnittstellen dieser Klasse stehen in OrgaDB.hpp
Preview.hpp Preview.cpp	5.000 67.888	Klassen zur Druckvorschau eines Abteilungs - Graphen
Print.hpp Print.cpp	2.965 19.564	Klasse zum Ausdrucken
PrintDlgs.hpp PrintDlgs.cpp	4.238 20.457	Dialogklassen zum Drucken
Propbox.hpp Propbox.cpp	6.924 28.988	kontextabhängiges Eigenschaftsfenster, ähnlich der Property - Box von Notes 4
PropWins.hpp PropWins.cpp	25.933 137.627	Darstellung der Eigenschaften graphischer Objekte in der Property - Box
Role.cpp	25.027	Verwaltung der Daten einer Rolle, Schnittstellen dieser Klasse stehen in OrgaDB.hpp
ScrllWin.hpp ScrllWin.cpp	3.508 15.609	Ableitung der Klasse Window, um Scrollbar- und Zoomfunktionalität erweitert, wird zur Darstellung von Graphen abgeleitet
StatusBr.hpp StatusBr.cpp	2.125 13.925	Statusleiste am unteren Rand des Applikationsfensters
SubMDI.hpp SubMDI.cpp	2.946 23.947	MDI - Fenster zur Darstellung von Tabellen oder Arbeitsgruppen einer Datenbank
Sysdepen.hpp Sysdepen.cpp	2.316 8.807	Systemspezifische Klassen, die pro Betriebssystem implementiert werden mußten (bisher OS/2 und Windows)
Trashcan.hpp Trashcan.cpp	3.069 24.220	graphischer "Mülleimer"
Unit.cpp	52.016	Verwaltung der Daten einer Abteilung, Schnittstellen dieser Klasse stehen in OrgaDB.hpp
UnitCtrl.hpp UnitCtrl.cpp	7.188 67.874	Klasse zur graphischen Darstellung und Benutzerinteraktion einer Abteilung
WGContrl.hpp WGContrl.cpp	7.915 79.667	Klasse zur graphischen Darstellung und Benutzerinteraktion einer Arbeitsgruppe
WGPrevWn.hpp WGPrevWn.cpp	4.131 46.432	Klassen zur Druckvorschau eines Arbeitsgruppen - Graphen
Workgrp.cpp	51.581	Verwaltung der Daten einer Arbeitsgruppe, Schnittstellen dieser Klasse stehen in OrgaDB.hpp
	gesamt: 2,408 MB	

Gesamter Quellcode (ohne Bilder): 2,718 Megabyte.

Anmerkung:

Die Idee zu den Klassen in den Dateien Cabinet2.cpp, Cabinet2.hpp, FixedBtn.cpp und FixedBtn.hpp wurde im Rahmen der Zusammenarbeit mehrerer Diplomarbeiten von **[Liebrand1995]** übernommen, die Klassen aber im äußeren Erscheinungsbild stark verändert und im Laufzeitverhalten optimiert.

Sonstige Dateien:

Bitmaps, Icons und *Mauszeiger* befinden sich in den Unterverzeichnissen 'gf-orga\bitmaps', 'gf-orga\icons' und 'gf-orga\pointer'. Auf eine Aufzählung dieser-zahl reichen Bilddateien soll hier verzichtet werden, da sie in der Ressourcedatei 'gf-orga.src' genau den jeweiligen Objekten zugeordnet werden.

Die folgenden Dateien werden z.T. beim Übersetzungsvorgang erstellt und stellen keinen eigenständigen Quellcode dar.

$McrWare.dll Macrware.dll	Makro - DLLs die Notes APIs benutzt, um Notes Datenbanken anzusprechen, jeweils für OS/2 und MS-Windows
GF-Orga.rc GF-Orga.res	Ressource - Dateien, die nach dem Übersetzungsvorgang der Datei GF-Orga.src durch den StarView - Ressourcecompiler entstehen
Template.nsf	leere Organisationsdatenbank, die kopiert wird, wenn eine neue Datenbank durch die Applikation erzeugt werden soll **Achtung:** die so kopierte Datenbank hat keine neue Replikations ID, so daß es zu Replikationskonflikten mit anderen Datenbanken kommen kann. In-nerhalb des Programmcodes wird eine neue Replikations ID erzeugt (Databses.cpp, class Databases.OpenEmptyOrgaDB()), jedoch hat die Macrware DLL noch nicht die Fähigkeit, diese neue ID in die Datenbank zu setzen. Aus diesem Grund ist der Menüpunkt 'Neue Datenbank erzeugen' voll implementiert, ist aber bis zum Erscheinen einer angepaßten Macrware DLL auskommentiert.

A.6.2 Klassenhierarchien

Es wurden insgesamt 108 mehr oder weniger umfangreiche Klassen implementiert. Kapitel A.6.2.1 zeigt direkt oder gar nicht abgeleitete Klassen. In Kapitel A.6.2.2 folgen Schaubilder mit den restlichen, indirekt abgeleiteten Klassen.
Die Hierarchie der StarView eigenen Klassen ist in **[StarView1992b]** nachzulesen.

A.6.2.1 Direkt von StarView abgeleitete Klassen

Die meisten Klassen sind direkt von StarView - Klassen abgeleitet und mit eigenen Attributen überladen worden. Die Überschrift nennt jeweils die StarView Klasse.

Klasse Control:

BrowserColumn	BrowserRow	Databases
FixedButton	Line	MoveControl

MyBrowseBox
MyPrint
OnOffField
OrganizationDB
PersonControl
PinInOutControl
UnitControl
WorkgroupControl

Klasse List:

SortedPersonCtrlList
SortedSubUnitList
UnitControlsList
WGControlsList

Klasse ModalDialog:

AboutBox CheckDialog DlgListBox
DlgPrintPreview MissingObjects ModalDialogDescription
ModalDialogUnknownObject ModalDialogYesNo NoHelpFileDlg
OpenDBDlg PastingDoubleObject PasteWithOrWithoutMembers
SaveInconsistentQuery

Klasse ModelessDialog:

BugReport DlgAbortPrint
DragDropLists PropertyBox

Klasse Window:

BackgroundColorWin BugReportPersons BugReportRoles
BugReportUnits BugReportWorkgroups Cabinet
FormatUnitGraph FormatWGGraph MyBrowserDataWin
PersonBasics PersonCoWorkers PersonResponsibleFor

PersonRoles PersonWGs PersonWorksFor
PreviewWin RoleBasicsWin RoleListBoxWin
RoleMLEWin ScrollableWin StatusBarWin
UnitBasics UnitListBoxWin UnitMLEWin

UnitsGraphWorkspace WGBackgroundColorWin WGGraphWorkspace
WGPreviewWin WindowAdressBooks WindowAutoSave
WindowCheckConsistency WindowDBBasics WindowPrinting
WindowStartup WindowTrashcan WorkgroupBasics
WorkgroupListBoxWin WorkgroupMLEWin

Klasse WorkWindow:

AppWin LoadWindow

sonstige:

DropableBitmap (FixedBitmap)
EditWithKeyInput (Edit)
Graphics (OutputDevice)
HorizontalScrollBar (ScrollBar)
ItemListBox (MultiListBox + Drag&Drop Funktionalität)
MultiLineEditWithKeyInput (MultiLineEdit)
MyApp (MDIApplication)
MyEdit (Edit, edit field with keyinput - handle)
OrgaMDI (MDIWindow)
SubMDI (MDIWindow)
Trashcan (FloatingWindow)
UnitsPool (FloatingWindow)
VerticalScrollBar (ScrollBar)

ohne Ableitung:

BrowseEvent CheckDB DragPool Functions
ML MyClipboard Person Role
Unit Utilities Workgroup

A.6.2.2 Klassenhierarchien der indirekt abgeleiteten Klassen

Ein Stern (*) kennzeichnet StarView - Klassen, die nicht im Rahmen dieser Arbeit entstanden sind:

DropEvent* BrowseEvent MouseEvent

BrowserDropEvent BrowserMouseEvent

Abbildung A5: Ableitung der Klassen BrowserDropEvent und BrowserMouseEvent

Window*

ScrollableWin

UnitGraph WorkgroupsGraph

Abbildung A6: Ableitung der Klassen für Abteilungs- und Arbeitsgruppengraphen

Control*

MyBrowseBox

OrgaBrowseBox

PersonsBrowser UnitsBrowser

RolesBrowser WorkgroupsBrowser

Abbildung A7: Ableitung der Browser - Klassen

Eidesstattliche Erklärung

Ich erkläre, daß ich die vorliegende Arbeit selbständig und ohne unerlaubte fremde Hilfe Dritter angefertigt, andere als die angegebenen Quellen und Hilfs mittel nicht benutzt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Paderborn, den 24.07.1996